

Traballo Fin de Grao

# HOMOLOXÍA PERSISTENTE EN 3-VARIEDADES

Pedro Tarrío Mallou

Xullo, 2024

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



GRADO DE MATEMÁTICAS

Trabajo Fin de Grado

# HOMOLOGÍA PERSISTENTE EN 3-VARIEDADES

Pedro Tarrío Mallou

Julio, 2024

UNIVERSIDAD DE SANTIAGO DE COMPOSTELA



# Trabajo propuesto

<b>Área de Conocimiento:</b> Geometría y Topología.
<b>Título:</b> Homología persistente en 3-variedades.
<b>Breve descripción del contenido</b>
Se estudiarán los fundamentos del análisis topológico de datos y de la geometría hiperbólica en variedades de dimensión 3. Cuenta con una parte experimental, con el fin de usar la homología persistente para obtener invariantes topológicos que distingan 3-variedades hiperbólicas.
<b>Recomendaciones</b>
Conocimiento básico en geometría y topología. A lo largo del trabajo, se citan numerosas referencias con fines introductorios a la materia.
<b>Otras observaciones</b>
Se usará el lenguaje de programación <code>Python</code> en la creación del código presentado en la parte experimental.



# Índice

<b>Resumen</b>	<b>VIII</b>
<b>Introducción</b>	<b>XI</b>
<b>1. Homología persistente</b>	<b>1</b>
1.1. Introducción al análisis topológico de datos . . . . .	1
1.2. Homología y filtraciones . . . . .	1
1.3. Objetos persistentes . . . . .	8
1.4. Diagramas de persistencia . . . . .	10
1.5. Landscapes . . . . .	18
1.6. Tratamiento estadístico . . . . .	21
<b>2. Geometría hiperbólica</b>	<b>23</b>
2.1. Variedades hiperbólicas y cubierta universal . . . . .	23
2.2. Aplicaciones cuasiconformes . . . . .	26
2.3. Teoría ergódica . . . . .	28
2.4. Teorema de Mostow . . . . .	31
<b>3. Homología persistente en 3-variedades hiperbólicas</b>	<b>41</b>
3.1. Variedades en SnapPy . . . . .	42
3.2. Muestreo aleatorio sobre una variedad . . . . .	43

---

3.3. Cálculo de landscapes . . . . .	48
3.4. Regiones de confianza mediante bootstrap . . . . .	52
3.5. Resultados . . . . .	54
<b>4. Conclusiones</b>	<b>57</b>
<b>I. Contrastes para algunas variedades</b>	<b>59</b>
<b>II. Borrador del programa</b>	<b>65</b>
<b>Bibliografía</b>	<b>87</b>





## Resumen

Se comienza con una introducción al análisis topológico de datos, presentando las herramientas básicas de esta disciplina: homología persistente, diagramas de persistencia y landscapes, teorema de estabilidad, entre otras. En particular, nos interesará la aplicación de la homología persistente sobre muestras finitas de puntos sobre una variedad, a partir de las construcciones de complejos simpliciales sobre dichas muestras, debido a su potencial para generar invariantes métricos.

Se sigue con los fundamentos de la geometría hiperbólica, estudiando algunos de los modelos clásicos del espacio hiperbólico, el cual es cubierta universal de cualquier variedad hiperbólica, particularizando el caso de dimensión 3. El objetivo principal de esta sección es demostrar el Teorema de Rigidez de Mostow en el caso compacto; para ello, se muestran previamente algunos conceptos y resultados usados en la demostración. Como consecuencia de este teorema, en el caso de 3-variedades hiperbólicas, la métrica es un invariante topológico y por lo tanto, variedades hiperbólicas no homeomorfas pueden ser distinguidas usando invariantes métricos como los que nos proporcionará la homología persistente.

Finalmente, se llevarán las técnicas anteriores a la práctica, con la realización de un programa que muestree puntos aleatorios sobre 3-variedades hiperbólicas compactas orientables, calcule los correspondientes diagramas de persistencia y landscapes, y compare mediante contrastes de hipótesis los resultados obtenidos para cualquier par de variedades hiperbólicas dadas, con el objetivo de distinguir las topológicamente con cierto grado de confianza.

## Abstract

We begin with an introduction to topological data analysis, presenting the basic tools of this discipline: persistent homology, persistence diagrams and landscapes, as well as the stability

theorem, among others. Specifically, we are interested in the application of persistent homology to finite samples of points on a manifold, using the construction of simplicial complexes on these samples, due to their potential to generate metric invariants.

Next, we cover the fundamentals of hyperbolic geometry, studying some of the classical models of hyperbolic space, which is the universal cover of any hyperbolic manifold, focusing on the 3-dimensional case. The main objective of this section is to demonstrate Mostow's Rigidity Theorem in the compact case. To this end, we present some concepts and results used in the proof. As a consequence of this theorem, in the case of 3-dimensional hyperbolic manifolds, the metric is a topological invariant. Therefore, non-homeomorphic hyperbolic manifolds can be distinguished using metric invariants, such as those provided by persistent homology.

Finally, we will put the previous techniques into practice by developing a program that samples random points on compact orientable 3-dimensional hyperbolic manifolds, calculates the corresponding persistence diagrams and landscapes, and compares the results obtained for any pair of given hyperbolic manifolds using hypothesis testing, with the aim of topologically distinguishing them with a certain degree of confidence.

# Introducción

El análisis topológico de datos se presenta como una potente herramienta matemática que hace uso de la topología algebraica. Se trata de una disciplina emergente en la actualidad con aplicaciones muy variadas en campos de ciencia aplicada, como neurociencia, finanzas o ciencia de materiales. En el presente trabajo, nos centraremos en un ámbito más puro de las matemáticas, concretamente, en su aplicación sobre variedades hiperbólicas compactas de dimensión 3.

En la clasificación de variedades hiperbólicas, es de especial interés el cálculo de invariantes topológicos que permitan distinguirlas. En el caso de dimensión 3, tenemos rigidez sobre la geometría de la variedad, de forma que dos variedades hiperbólicas de volumen finito homeomorfas son, de hecho, isométricas. Esto nos permite afirmar que los invariantes métricos serán también topológicos, como el volumen de la variedad. Luego utilizaremos el análisis topológico de datos, concretamente la homología persistente, para obtener dichos invariantes, que se estimarán estadísticamente. Nos restringiremos al caso de las 3-variedades hiperbólicas compactas orientables utilizando SnapPy [1], un programa disponible como módulo en Python destinado al estudio de la geometría hiperbólica, y Ripser [2], paquete para el cálculo de homología persistente. Como gran motivación de este trabajo, se intenta encontrar invariantes topológicos que permitan distinguir variedades que SnapPy no consiga diferenciar, que corresponde principalmente a las variedades compactas orientables de mismo volumen y homología.

El trabajo está estructurado en cuatro capítulos, los dos primeros de carácter teórico e introductorio a las técnicas utilizadas en la parte experimental. Se comienza con una breve introducción al análisis topológico de datos, dando las definiciones básicas sobre homología simplicial. Se presenta el formalismo detrás de los diagramas de persistencia, los cuales se tratan de invariantes métricos, y demostrando el teorema de estabilidad que justifica su uso en la práctica. A partir de ellos, se definen los landscapes, para los cuales se explica el tratamiento estadístico necesario para trabajar con muestras aleatorias en espacios métricos.

En el capítulo 2, se estudian los fundamentos de la geometría hiperbólica con el objetivo de familiarizarse con la métrica del espacio hiperbólico y algunos de los modelos más relevantes del mismo. La rigidez de la geometría hiperbólica para dimensión mayor o igual a 3 será fundamental

para la parte experimental de este trabajo; esto es consecuencia del teorema principal presentado en esta memoria, el Teorema de Rigidez de Mostow, el cual se prueba con bastante detalle para el caso de variedades hiperbólicas compactas.

A continuación, asentados todos los aspectos teóricos, se pasa a explicar la parte computacional, la cual consiste en la creación de un programa para distinguir variedades hiperbólicas compactas orientables de dimensión tres. Para ello, este debe ser capaz de muestrear puntos de forma aleatoria sobre una variedad dada, calcular el diagrama de persistencia y landscape asociado, y dar regiones de confianza para el landscape de la variedad de donde fue tomada la muestra a partir de los cálculos anteriores. Todo con el fin de realizar contrastes de hipótesis entre variedades para determinar con cierto nivel de confianza si son o no homeomorfas.

Finalmente, se concluye con una breve recopilación de los principales objetivos alcanzados, así como la presentación de posibles mejoras en el programa y potenciales líneas de investigación futuras relacionadas con el trabajo.

# Capítulo 1

## Homología persistente

### 1.1. Introducción al análisis topológico de datos

Una de las principales motivaciones de los analistas de datos, es lograr distinguir y clasificar conjuntos de datos mediante características calculables computacionalmente. El TDA (*Topological data analysis*) hace uso de la topología algebraica para extraer invariantes topológicos de enormes nubes de puntos. En general, nuestro problema comienza con una muestra finita de puntos  $\mathbb{X}$ , que suponemos procedente de un espacio métrico  $X$  desconocido. El objetivo último será recuperar la información de la topología y métrica de  $X$  con cierto nivel de confianza a partir del análisis computacional de la muestra.

En este capítulo, introduciremos al lector los conceptos necesarios para comprender el uso de esta herramienta, desde las definiciones básicas de álgebra homológica, hasta resultados que nos garanticen la efectividad de estos métodos en la práctica, siguiendo las referencias [3], [4], [5] y [6], principalmente.

### 1.2. Homología y filtraciones

Buscamos invariantes topológicos que puedan ser calculables de forma eficiente, el invariante elegido será la homología. Nos referimos a [7] para una presentación más general e introductoria a los grupos de homología. En nuestro caso, estaremos interesados en la homología simplicial y singular como mecanismo para calcular los invariantes deseados, que introducimos a continuación.

**Definición 1.1.** Un *complejo simplicial (abstracto)* es un par  $(V, \Sigma)$ , donde  $V$  es un conjunto finito y  $\Sigma$  una familia no vacía de subconjuntos de  $V$  tal que, dado  $\sigma \in \Sigma$  y  $\tau \subseteq \sigma$ , entonces se tiene  $\tau \in \Sigma$ .

Llamaremos a  $V$  conjunto de vértices y  $\Sigma$  conjunto de símlices. Además, se dirá que  $\sigma$  es un  $k$ -símlice si tiene  $k + 1$  vértices, denotando como  $\Sigma_k$  al conjunto de  $k$ -símlices del complejo simplicial. De esta forma, se puede entender un complejo simplicial como la unión de: puntos (0-símlices), segmentos (1-símlices), triángulos (2-símlices), tetraedros (3-símlices) y análogos de dimensión mayor. Cuando se represente gráficamente un complejo simplicial, rellenaremos con color el interior de los  $k$ -símlices para diferenciarlos de la unión de  $k+1$  ( $k - 1$ )-símlices. Por ejemplo, en la figura 1.1 podemos diferenciar un 2-símlice (triángulo gris) de lo que serían simplemente tres 1-símlices que forman un triángulo no coloreado.

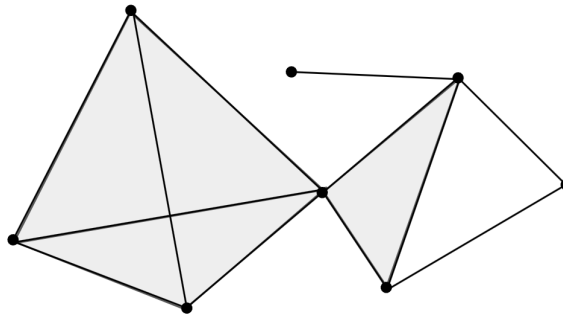


Figura 1.1: Complejo simplicial con 8 vértices, 11 segmentos y 5 triángulos (3D).

*Observación 1.2.* Podemos dotar a un complejo simplicial de estructura de espacio topológico mediante la biyección  $\phi : V \rightarrow \{1, 2, \dots, N\}$ , donde  $N$  es el número de vértices, de forma que generamos un subespacio de  $\mathbb{R}^N$  dado por  $\bigcup_{\sigma \in \Sigma} c(\sigma)$ , siendo  $c(\sigma)$  la envolvente convexa del conjunto de vectores de la base canónica  $\{e_{\phi(v)}\}_{v \in \sigma}$ .

**Definición 1.3.** Diremos que un espacio topológico  $X$  es *triangulable* si existe un complejo simplicial con espacio topológico subyacente homeomorfo a  $X$ .

Dotado  $V$  de orden total, se define el  $i$ -ésimo operador cara  $d_i : \Sigma_k \rightarrow \Sigma_{k-1}$  tal que  $d_i(\sigma) = \sigma - \{v_i\}$ , donde  $v_i$  es el  $i$ -ésimo vértice en  $\sigma$ , con  $i \in \mathbb{N}_0$  entre 0 y  $k$  incluidos. Dado un complejo simplicial  $\Delta = (V, \Sigma)$  y  $A$  un anillo abeliano, definimos el grupo de  $k$ -cadenas  $C_k(\Delta, A)$  como el módulo libre con coeficientes en  $A$  generado por los  $k$ -símlices de  $\Sigma_k$  con coeficientes en  $A$ . Podemos considerar la orientación de un símlice a partir de un orden establecido en su conjunto de vértices, decidiendo el sentido en el que se “recorren” los vértices, teniendo que para cada símlice orientado  $(v_0, v_1, \dots, v_{k-1}, v_k) = -(v_k, v_{k-1}, \dots, v_1, v_0)$ . Cualquier trasposición de dos vértices cambia la orientación, por lo tanto cada símlice admite únicamente dos orientaciones; así, dos símlices ordenados con el mismo número de vértices tendrán la misma orientación si y solo si difieren en una permutación par.

**Definición 1.4.** Operador borde  $\partial_k : C_k(\Delta) \rightarrow C_{k-1}(\Delta)$ :

$$\partial_k = \sum_{i=0}^k (-1)^i d_i,$$

donde se considera la orientación inducida en el operador cara. Diremos que una  $k$ -cadena  $c \in C_k(\Delta)$  es un  $k$ -ciclo si  $\partial_k(c) = 0$ . Una  $k$ -cadena  $c \in C_k(\Delta)$ , es un  $k$ -borde si existe una  $k+1$ -cadena  $\rho$  tal que  $\partial_{k+1}(\rho) = c$ .

**Proposición 1.5.** El borde de cualquier  $k$ -cadena no tiene borde ( $k \geq 1$ ), i.e.,  $\partial^2 \equiv \partial_{k-1} \circ \partial_k \equiv 0$ .

*Demostración.* Basta con probar que es cierto para los  $k$ -símplices, y automáticamente se tiene el resultado para las  $k$ -cadenas por expresarse como combinación lineal de ellos. Sea  $\sigma \in \Sigma_k$ :

$$\begin{aligned} \partial_{k-1}(\partial_k(\sigma)) &= \partial_{k-1} \left( \sum_{i=0}^k (-1)^i d_i(\sigma) \right) = \sum_{i=0}^k (-1)^i \partial_{k-1}(d_i(\sigma)) \\ &= \sum_{i=0}^k (-1)^i \sum_{\substack{j=0 \\ j < i}}^{k-1} (-1)^j d_j(d_i(\sigma)) = \sum_{\substack{i=1 \\ j < i}}^k (-1)^{i+j} d_j(d_i(\sigma)) + \sum_{\substack{i=0 \\ i \leq j}}^{k-1} (-1)^{i+j} d_j(d_i(\sigma)) \\ &= \sum_{\substack{i=1 \\ j < i}}^k (-1)^{i+j} d_{i-1}(d_j(\sigma)) + \sum_{\substack{i=0 \\ i \leq j}}^{k-1} (-1)^{i+j} d_j(d_i(\sigma)) \\ &\stackrel{i' \equiv i-1}{=} \sum_{\substack{j=0 \\ j \leq i'}}^{k-1} (-1)^{i'+j+1} d_{i'}(d_j(\sigma)) + \sum_{\substack{i=0 \\ i \leq j}}^{k-1} (-1)^{i+j} d_j(d_i(\sigma)) \\ &\stackrel{i \equiv i', i \leftrightarrow j}{=} \sum_{\substack{i=0 \\ i \leq j}}^{k-1} (-1)^{i+j+1} d_j(d_i(\sigma)) + \sum_{\substack{i=0 \\ i \leq j}}^{k-1} (-1)^{i+j} d_j(d_i(\sigma)) = 0 \quad \square \end{aligned}$$

**Corolario 1.6.** Del resultado anterior se sigue que  $\text{Img}(\partial_{k+1}) \subseteq \text{Ker}(\partial_k)$ .

Por ahora, vemos que los complejos simpliciales y los operadores lineales definidos tienen una gran facilidad para ser tratados computacionalmente, que recordemos que es lo buscado. A partir de estos conceptos, para mantener el poder de cálculo, podemos definir la homología simplicial.

**Definición 1.7.** El grupo de homología simplicial de grado  $k$  con coeficientes en un anillo abeliano  $A$  de un complejo simplicial  $\Delta$ , denotado por  $H_k(\Delta, A)$ , viene dado por:

$$H_k(\Delta, A) \cong \text{Ker}(\partial_k) / \text{Img}(\partial_{k+1}).$$

Definimos el  $k$ -ésimo número de Betti como  $\beta_k = \text{Rango } H_k(\Delta, A)$ . Si  $A$  es un cuerpo, el grupo de homología es un espacio vectorial sobre  $A$  y el número de Betti es su dimensión.

**Definición 1.8.** Sean  $\Delta_1, \Delta_2$  dos complejos simpliciales. Una *aplicación simplicial*  $f$  entre  $\Delta_1$  y  $\Delta_2$  es una función de  $V(\Delta_1)$  en  $V(\Delta_2)$  tal que la imagen de un símplice de  $\Delta_1$  es un símplice de  $\Delta_2$ , entendiendo el símplice como un subconjunto del conjunto de vértices.

*Observación 1.9.* Una aplicación simplicial  $f : \Delta_1 \rightarrow \Delta_2$  induce un homomorfismo entre los grupos de homología  $f_k : H_k(\Delta_1) \rightarrow H_k(\Delta_2)$  para todo entero no negativo  $k$ , definido por

$$f_k([c]) = [f'(c)],$$

donde  $f'$  es la aplicación inducida en los grupos de cadenas, definido en los símplices como  $f'([v_0, \dots, v_n]) = [f(v_0), \dots, f(v_n)]$  y extendido linealmente a las cadenas;  $[c] \in H_k(\Delta_1)$  es una clase de homología representada por un ciclo  $c$  en  $\Delta_1$  y  $[f'(c)] \in H_k(\Delta_2)$  es la clase de homología representada por el ciclo  $f(c)$  en  $\Delta_2$ .

Se puede ver que  $f'$  conmuta con el operador borde, por tanto lleva  $k$ -ciclos de  $\Delta_1$  en  $k$ -ciclos de  $\Delta_2$  y la imagen por  $f'$  de una cadena cobordante es también cobordante. En consecuencia, el morfismo a nivel de cadenas simpliciales define en el cociente un morfismo entre los grupos de homología.

A continuación presentamos algunas definiciones básicas de homología singular necesarias para la comprensión de un teorema posterior de la sección 1.4, se trata de un concepto más general ya que se define sobre espacios topológicos arbitrarios; se recomienda las referencias [7] y [8] para mayor detalle. En la práctica, se utilizará homología simplicial ya que trabajaremos sobre nubes de puntos finitas.

**Definición 1.10.** Sean  $\{x_0, \dots, x_n\} \in \mathbb{R}^m$  geoméricamente independientes. Se llama  *$n$ -símplice geométrico*  $\tilde{\sigma}$  generado por  $\{x_0, \dots, x_n\} \in \mathbb{R}^m$ , al conjunto de puntos

$$x = \sum_{i=0}^n t_i x_i,$$

donde  $\sum_{i=0}^n t_i = 1$  y  $0 \leq t_i \leq 1 \forall i \in \{0, \dots, n\}$ . En particular, sea  $e_i = (0, 0, \dots, 1, 0, 0, \dots)$  la base canónica de  $\mathbb{R}^\infty$ . Se llama  *$n$ -símplice estándar*  $\tilde{\Delta}^n \subset \mathbb{R}^{n+1} \subset \mathbb{R}^\infty$  al  $n$ -símplice geométrico generado por  $\{e_0, e_1, \dots, e_n\}$ ; que es precisamente el espacio subyacente de un símplice, caso particular de la Observación 1.2.

En lo que sigue, sea  $X$  un espacio topológico y  $A$  un anillo conmutativo.

**Definición 1.11.** Un  *$n$ -símplice singular de  $X$*  es una aplicación continua

$$\tau : \tilde{\Delta}^n \rightarrow X.$$

**Definición 1.12.** Una  *$n$ -cadena singular*  $\tilde{c}$  es una combinación lineal finita

$$\tilde{c} = \sum_i a_i \tau_i$$

con  $a_i \in A$  (nulos excepto un número finito) y  $\tau_i$   $n$ -símplices singulares. El  $n$ -ésimo grupo de cadenas singulares  $S_n(X)$  en  $A$  es el grupo abeliano libre generado por los  $n$ -símplices singulares.

Surgen los operadores cara y borde de la siguiente manera,

**Definición 1.13.** El  $i$ -ésimo operador cara  $\tilde{d}_i : S_n(X) \rightarrow S_{n-1}(X)$  actúa sobre un  $n$ -símplice singular como:

$$\begin{aligned} \tilde{d}_i(\tau) : \tilde{\Delta}^{n-1} &\rightarrow X \\ (t_0, \dots, t_{i-1}, t_i, \dots, t_{n-1}) &\mapsto \tau(t_0, \dots, t_{i-1}, 0, t_i, \dots, t_{n-1}). \end{aligned}$$

Por linealidad,  $\tilde{d}_i$  se extiende a  $S_n(X)$ , es decir,  $\tilde{d}_i(\sum_i a_i \tau_i) = \sum_i a_i \tilde{d}_i(\tau_i)$ .

**Definición 1.14.** El  $n$ -ésimo operador borde  $\tilde{\partial}_n : S_n(X) \rightarrow S_{n-1}(X)$  viene dado por

$$\tilde{\partial}_n = \sum_{i=0}^n (-1)^i \tilde{d}_i,$$

donde  $\tilde{d}_i$  es el  $i$ -ésimo operador cara. La noción de ciclo y borde se define de forma análoga al caso simplicial.

Véase que el resultado de la proposición 1.5 también se tiene aquí,  $\tilde{\partial}_{n-1} \circ \tilde{\partial}_n \equiv 0$  para  $n \geq 2$ , y la demostración es idéntica. De igual manera, se obtiene que  $\text{Img}(\tilde{\partial}_{n+1}) \subseteq \text{Ker}(\tilde{\partial}_n)$ .

**Definición 1.15.** El  $n$ -ésimo grupo de homología singular de  $X$  es

$$H_n(X) = \text{Ker}(\tilde{\partial}_n) / \text{Img}(\tilde{\partial}_{n+1}).$$

El número de Betti  $\beta_n$  será el rango del grupo  $H_n(X)$ .

Sea ahora  $f : X \rightarrow Y$  una aplicación continua entre espacios topológicos, se tiene la siguiente interpretación entre grupos de  $n$ -cadenas singulares  $f^\# : S_n(X) \rightarrow S_n(Y)$  definida por  $f^\#(\sum_i a_i \tau_i) = \sum_i a_i f^\#(\tau) = \sum_i a_i (f \circ \tau)$ . Es fácil ver que  $f^\#$  conmuta con el operador borde y por tanto induce un homomorfismo entre los grupos de homología:

$$\begin{aligned} f_* : H_n(X) &\rightarrow H_n(Y) \\ [\xi] &\mapsto f_*([\xi]) = [f^\#(\xi)], \end{aligned}$$

siendo  $[\xi]$  una clase de homología con representante el  $n$ -ciclo  $\xi$ . Sea  $g : X \rightarrow Y$  otra aplicación continua, si  $f$  y  $g$  son homotópicas, entonces inducen el mismo homomorfismo en homología  $f_* = g_*$ .

Una equivalencia de homotopía entre espacios topológicos induce un isomorfismo a nivel de homología singular, luego dos espacios homotópicamente equivalentes tienen los mismos grupos

de homología. En [9] se realizan los cálculos detallados que nos permiten afirmar que la homología simplicial de un complejo es isomorfa a la homología singular de su espacio topológico asociado definido en la Observación 1.2, como consecuencia, la homología simplicial de un espacio topológico triangulable no depende de la descomposición simplicial escogida.

Nuestro objetivo ahora es definir un método para construir complejos a partir de nubes de puntos, que consigan aproximar eficientemente la forma y topología del espacio del que fueron muestreados. De esta forma, podremos estimar la homología de un espacio  $X$  mediante el cálculo de la homología de su complejo simplicial asociado. Una forma de asegurarse de que el resultado es correcto, será encontrar una equivalencia de homotopía de  $X$  al complejo.

**Definición 1.16.** Sea  $X$  un espacio topológico y  $\mathcal{U} = \{U_v\}_{v \in W}$  un recubrimiento finito de  $X$ , se define el *nervio* de  $\mathcal{U}$ , que escribiremos como  $N(\mathcal{U})$ , al complejo simplicial con conjunto de vértices  $W$  tal que los vértices  $\{v_0, \dots, v_k\}$  generan un  $k$ -símplice si y solo si  $\bigcap_{i=0}^k U_{v_i} \neq \emptyset$ .

En [10], se enuncia y demuestra el *teorema del nervio*, en particular, nos garantiza que si  $\mathcal{U}$  es un recubrimiento abierto finito y para todo subconjunto  $\emptyset \neq V \subseteq W$  se cumple que  $\bigcap_{v \in V} U_v$  es o bien vacío o bien homótopo a un punto, entonces  $N(\mathcal{U})$  es homotópicamente equivalente a  $X$ . De este modo, nuestro problema se reduce simplemente a generar recubrimientos bajo estas condiciones. A continuación se presentan algunos métodos aplicados sobre espacios métricos, que serán los de especial interés para la parte experimental de este trabajo.

**Definición 1.17.** Sea  $X$  un espacio métrico y consideremos que existe un recubrimiento finito de bolas abiertas de radio  $\epsilon$  con centros en  $v \in V$ ,  $\{B_\epsilon(v)\}_{v \in V}$ . Llamaremos *complejo de Čech asociado a  $V$  y  $\epsilon$* , denotado por  $\check{C}(V, \epsilon)$ , al nervio del recubrimiento anterior.

La observación clave de esta construcción, es que si  $X$  es una variedad Riemanniana compacta, tenemos garantizada la existencia de  $r$  positivo y  $V$  subconjunto finito de  $X$  tal que  $\forall \epsilon \leq r$   $\check{C}(V, \epsilon)$  es homotópicamente equivalente a  $X$ .

**Definición 1.18.** Sea  $X$  un espacio métrico finito, el *complejo de Vietoris-Rips asociado a  $X$  y  $\epsilon$* , denotado por  $VR(X, \epsilon)$ , viene dado por el complejo simplicial de vértices  $X$  tal que  $\{x_0, \dots, x_k\}$  generan un  $k$ -símplice si y solo si  $d(x_i, x_j) \leq \epsilon \quad \forall i, j \in [0, k]$ .

En la figura 1.2, se muestra gráficamente la construcción del complejo de Vietoris-Rips asociado al parámetro  $\epsilon$  para tres valores distintos. La distancia  $\epsilon$  está directamente relacionada con el radio de las bolas  $R$  mediante  $\epsilon=2R$ , de forma que varios vértices formarán un símplice si para cada vértice, su bola interseca la bola de cada vértice restante; o lo que es lo mismo, si para cada vértice, la distancia al resto de vértices es menor o igual a  $\epsilon$ . Para el primer radio  $R_1$ , solo se forman 0-símplices. A medida que aumentamos el radio, las bolas comienzan a intersecarse formando símplices de mayor dimensión. Se tiene la siguiente relación entre los complejos anteriores,

$$\check{C}(X, \epsilon) \subseteq VR(X, 2\epsilon) \subseteq \check{C}(X, 2\epsilon),$$

la cual nos garantiza poder usar la construcción de Vietoris-Rips sin perder información relevante en el cálculo homológico.

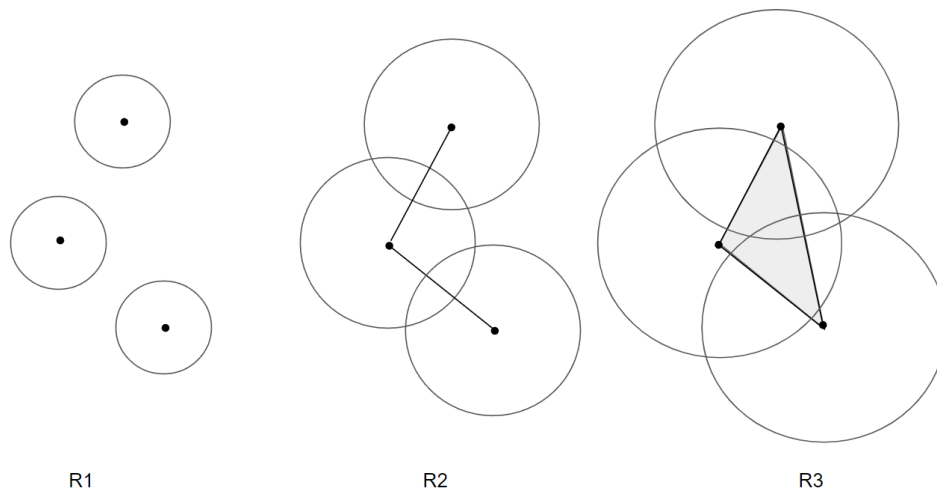


Figura 1.2: Construcción del complejo de Vietoris-Rips para tres radios sobre un espacio métrico de tres puntos.

En el complejo de Vietoris-Rips, el conjunto de vértices es el propio espacio métrico, lo cual implica que el coste computacional será alto para muestras considerablemente grandes. Existen varias soluciones ante este problema. Una de ellas consiste en considerar un subconjunto finito más pequeño  $\mathcal{R}$  del espacio métrico original  $X$ . Sus elementos son los llamados *puntos de referencia*,  $\lambda \in \mathcal{R}$ . A partir de ellos, se divide el espacio métrico en las celdas de Voronoi, que para cada punto de referencia viene dada por:

$$V_\lambda = \{x \in X \mid d(x, \lambda) \leq d(x, \lambda') \quad \forall \lambda' \in \mathcal{R}\}.$$

Estas celdas de Voronoi generan un recubrimiento de  $X$  a partir del cual podemos considerar su nervio, que definimos como el *complejo de Delaunay asociado a  $\mathcal{R}$* . El problema de esta definición, es que para muestras finitas, será poco probable encontrar puntos situados a la misma distancia de dos o más puntos de referencia, luego casi todos los símlices generados serán discretos. Para solucionar esto, tendríamos que ser algo más flexibles en la frontera de las celdas, controlando mediante un parámetro adicional cómo de permisivos seremos en las proximidades de las fronteras. Existen diversas construcciones de este tipo, como el complejo de testigo fuerte y débil en [11].

Todas las construcciones presentadas tienen en común su dependencia con  $\epsilon$ . Aumentar este parámetro solo sirve para crear nuevos símlices, no destruye lo ya existente; una familia uniparamétrica de complejos que cumpla lo anterior se llama *filtración*. En la práctica, usaremos los complejos de Vietoris-Rips, que dan lugar a la filtración de Rips.

Ahora bien, ¿cómo sabemos cuál es el valor ideal del parámetro que genera el complejo simplicial más representativo de nuestro espacio original? La idea será obtener una filtración del espacio sobre el que aplicar técnicas de homología simplicial para obtener características topológicas para cada valor de  $\epsilon$ . Al analizar estas características en conjunto, aquellas que persistan durante un intervalo de parámetros de la filtración mayor serán consideradas como más representativas del conjunto de datos. Este concepto de persistencia se formaliza en la siguiente sección.

### 1.3. Objetos persistentes

Para introducirse a la teoría de categorías, funtores y transformaciones naturales se recomienda la referencia [12]. Sea ahora  $\mathcal{C}$  una categoría y  $P$  un conjunto parcialmente ordenado. Definamos la categoría  $\mathcal{P}$  con conjunto de objetos  $P$ , tal que dados  $x, y \in P$  se establece un único morfismo de  $x$  a  $y$  si  $x \leq y$ .

**Definición 1.19.** Un *objeto  $P$ -persistente en  $\mathcal{C}$*  es un funtor  $\Phi : \mathcal{P} \rightarrow \mathcal{C}$ , es decir, una familia de objetos  $\{c_x\}_{x \in P}$  de la categoría  $\mathcal{C}$  con morfismos  $\phi_{xy} : c_x \rightarrow c_y$  si  $x \leq y$ , que cumplen  $\phi_{zy} \circ \phi_{xz} = \phi_{xy}$  cuando  $x \leq z \leq y$ .

Además, el conjunto de objetos  $P$ -persistentes en  $\mathcal{C}$  forman en si mismo una categoría, que denotaremos por  $P_{pers}(\mathcal{C})$ , donde un morfismo entre funtores  $\Phi$  y  $\Psi$  es una transformación natural, entendida como una familia de morfismos entre la familia  $\{c_x, \phi_{xy}\}$  y  $\{d_x, \psi_{xy}\}$ . Ahora, sea  $f : P \rightarrow Q$  una aplicación que preserve el orden parcial entre conjuntos parcialmente ordenados, se obtiene el funtor  $f^* : P_{pers}(\mathcal{C}) \rightarrow Q_{pers}(\mathcal{C})$  dado por  $f^*(\Psi) = \Psi \circ F$ , siendo  $F$  la aplicación  $f$  entendida como funtor entre  $\mathcal{P}$  y  $\mathcal{Q}$ .

Véase que las construcciones de complejos de la sección anterior dan lugar a objetos  $\mathbb{R}$ -persistentes, pudiendo construir entonces los grupos de homología asociados y entenderlos como grupos  $\mathbb{R}$ -persistentes, e igual para los complejos de cadenas. El problema es que no disponemos de un teorema de clasificación de grupos abelianos  $\mathbb{R}$ -persistentes con el que estudiar y diferenciar directamente los objetos anteriores, luego tendremos que realizar algunas simplificaciones. En el Teorema 3.1 de la referencia [13] se presenta una clasificación para  $K$ -espacios vectoriales  $\mathbb{N}$ -persistentes que cumplan ciertas condiciones, a continuación presentamos algunas de las ideas claves para llegar a ello.

**Definición 1.20.** Sea  $\{G_n\}$  un grupo abeliano  $\mathbb{N}$ -persistente. El módulo graduado asociado  $\Theta(\{G_n\})$  sobre el anillo de polinomios graduado  $\mathbb{Z}[t]$  viene dado por

$$\Theta(\{G_n\}) = \bigoplus_{s \geq 0} G_s,$$

donde el generador de polinomios  $t$  actúa sobre los objetos de la categoría  $\{\alpha_n\}$  como

$$t \cdot \{\alpha_n\} = \{\psi_{n-1,n}(\alpha_{n-1})\},$$

con  $\psi_{xy}$  morfismo entre objetos de la categoría.

De la definición anterior concluimos que la categoría de grupos abelianos  $\mathbb{N}$ -persistentes es equivalente a la categoría de módulos no negativamente graduados sobre  $\mathbb{Z}[t]$  que pueden ser infinitamente generados.

**Definición 1.21.** Diremos que un  $K$ -espacio vectorial  $\mathbb{N}$ -persistente  $\{V_n\}_n$  es “tame” si cada espacio vectorial  $V_n$  es de dimensión finita y  $\psi_{n-1,n} : V_n \rightarrow V_{n+1}$  es un isomorfismo para  $n$  suficientemente grande.

**Teorema 1.22** (Gabriel [14], enunciado de [3]).  $\{V_n\}_n$  es “tame” si y solo si  $\Theta(\{V_n\}_n)$  es un  $K[t]$ -módulo graduado finitamente generado. Además, este puede descomponerse como

$$\{V_n\}_n \cong \bigoplus_{i=0}^N U(b_i, d_i)$$

donde  $b_i$  es un entero no negativo, igual que  $d_i$  salvo que este también puede tomar el valor  $\infty$ . Donde  $U(b, d)$  es un  $K$ -espacio vectorial  $\mathbb{N}$ -persistente definido como  $U(b, d)_t = K$  si  $b \leq t \leq d$ , con  $\psi_{s,t} = Id_K$  si  $b \leq s \leq t \leq d$ , y  $U(b, d)_t = 0$  en el resto. Esta descomposición es única en los pares  $(b_i, d_i)$  salvo el orden en los factores.

A los intervalos  $(b, d)$  será a lo que llamemos *código de barras*. Recapitulando, suponiendo que disponemos de una muestra de puntos finita  $\mathbb{X}$  de un espacio métrico  $X$ , los pasos a seguir para estudiar su homología a partir de la de los complejos simpliciales construidos será la siguiente.

- Construir el complejo simplicial  $\mathbb{R}$ -persistente con alguno de los métodos explicados.
- Establecer una aplicación que preserve el orden de  $\mathbb{N}$  en  $\mathbb{R}$ .
- A partir de esta aplicación, construir el complejo simplicial  $\mathbb{N}$ -persistente (filtración) asociado. Los morfismos en esta filtración vienen dados por la inclusión.
- Construir el complejo de cadenas  $\mathbb{N}$ -persistente denotado por  $\{C_*(n)\}_n$  con coeficientes en el cuerpo  $K$ , donde los morfismos serán las aplicaciones simpliciales inducidas por la inclusión.

- Calcular los códigos de barras asociados al  $K$ -espacio vectorial  $\mathbb{N}$ -persistente dado por  $\{H_i(C_*(n), K)\}_n$  donde, de nuevo, los morfismos en este espacio vectorial son las aplicaciones inducidas en homología por la inclusión.

La aplicación del segundo punto dará una sucesión de parámetros donde hay creación de simplices en la filtración. Puesto que trabajaremos con una nube de puntos finita, la sucesión se estabilizará y el módulo asociado será tame.

Observando el código de barras se puede extraer la información y propiedades de la homología de la muestra  $\mathbb{X}$ . Dispondremos de barras que nacen en  $b$  y mueren en  $d$ , siendo esta escala temporal la marcada por la variación del parámetro  $\epsilon$ . Cada barra representa una clase de homología, así podremos identificar fácilmente cuales son las clases que persisten durante más tiempo, que, en general, se espera que tengan mayor relación con las clases reales del espacio  $X$  del que se tomó la muestra.

## 1.4. Diagramas de persistencia

En esta sección, introducimos el diagrama de persistencia de una función real sobre un complejo simplicial, que se trata de un multiconjunto de puntos en el plano extendido  $\overline{\mathbb{R}}^2 = \overline{\mathbb{R}} \times \overline{\mathbb{R}}$ , con  $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ . Un multiconjunto es una generalización del concepto de conjunto, donde cada elemento cuenta con multiplicidad, que indica el número de veces que aparece. En relación a la sección anterior, el diagrama de persistencia se definirá como el multiconjunto de pares  $(b, d)$ . Principalmente, daremos resultados de estabilidad sobre estos diagramas; esto es, garantizaremos que los diagramas no varían fuertemente bajo pequeñas perturbaciones en el conjunto de datos. Así, en la práctica, nos aseguraremos que los diagramas de dos muestras ligeramente distintas de un mismo espacio serán similares.

De aquí en adelante, asumiremos que  $X$  es un espacio topológico triangulable. Fijaremos un cuerpo  $K$ , denotando el grupo de homología de grado  $k$  con coeficientes en  $K$  de un espacio topológico  $X$  como  $H_k(X)$ , que en este caso será un espacio vectorial.

**Definición 1.23.** Sea  $f$  una función real continua en  $X$ . Un *valor crítico de homología* de  $f$  es un número real  $C$  para el cual hay al menos un grado de homología  $k$  de forma que  $\exists \epsilon' > 0$  tal que  $\forall 0 < \epsilon \leq \epsilon'$  la aplicación inducida en homología singular por la inclusión  $H_k(f^{-1}(-\infty, C - \epsilon]) \rightarrow H_k(f^{-1}(-\infty, C + \epsilon])$  no es un isomorfismo.

**Ejemplo 1.24.** Consideremos como espacio topológico la figura del ocho dotada de la topología usual de  $\mathbb{R}^2$  restringida a ese subconjunto. Sea  $f$  una función real que asocia a cada punto la proyección vertical como se observa en la figura 1.3, considerando que los valores crecen en el

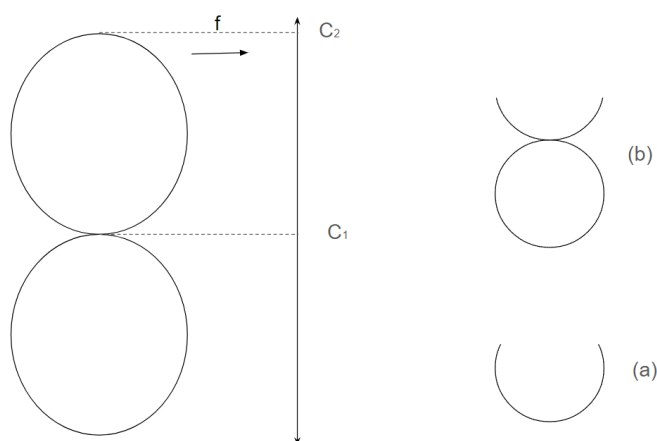


Figura 1.3: Figura del ocho como espacio topológico con una función real  $f$ . Junto a los conjuntos de subnivel de  $C_1 - \epsilon$  (a) y  $C_1 + \epsilon$  (b).

sentido hacia arriba. Los números reales  $C_1$  y  $C_2$  son valores críticos de homología singular de  $f$ . En efecto, si consideramos los grupos de homología singular de grado 1, que corresponden al abelianizado del grupo fundamental, tenemos:

- Para  $C_1$ , para todos los  $\epsilon$  a partir de uno suficientemente pequeño,  $f^{-1}(-\infty, C_1 - \epsilon]$ , correspondiente a la figura con la etiqueta (a), es contráctil y por lo tanto,  $\beta_1 = 0$ . En cambio,  $f^{-1}(-\infty, C_1 + \epsilon]$  representado con la etiqueta (b), es homeomorfo a  $S^1$ , luego  $\beta_1 = 1$ . Luego tenemos garantizado que la aplicación inducida por la inclusión no es un isomorfismo.
- Análogamente, con  $C_2$ , pasaríamos de un espacio homeomorfo a (b) de  $\beta_1 = 1$  a la figura del ocho con  $\beta_1 = 2$ , llegando al mismo resultado.

En palabras, los valores críticos de homología son los niveles para los cuales la homología de los conjuntos de subnivel cambia. A continuación, volvemos a introducir el concepto de “tame”, en este caso para funciones reales sobre un espacio topológico.

**Definición 1.25.** Diremos que  $f : X \rightarrow \mathbb{R}$  es “tame” si el número de valores críticos de homología es finito y  $\dim H_k(f^{-1}(-\infty, x))$  es finita  $\forall k \in \mathbb{Z}, \forall x \in \mathbb{R}$ .

En particular, las funciones de Morse sobre variedades compactas y las funciones lineales a trozos sobre complejos simpliciales finitos son “tame”. De aquí en adelante, usaremos en esta sección la notación dada por  $F_x = H_k(f^{-1}(-\infty, x))$  cuando se haya fijado  $k$ , y denotaremos por  $f_x^y : F_x \rightarrow F_y$  a la aplicación inducida por la inclusión del conjunto de subnivel de  $x$  en el de  $y$ , con  $x \leq y$ .

**Proposición 1.26.** *Dado un intervalo  $[x, y]$  que no contiene valores críticos de homología, se tiene que  $f_x^y$  es un isomorfismo para todo grado  $k$ .*

*Demostración.* Procedemos por reducción al absurdo, supongamos que  $f_x^y$  no es un isomorfismo para algún  $k$ . Luego sea  $z$  el punto medio entre  $x$  e  $y$ , tendríamos que para  $f_x^z$  y  $f_z^y$ , al menos uno tampoco es un isomorfismo. Realizando este procedimiento, de dividir los intervalos a la mitad y escoger uno para el que no se tiene un isomorfismo, infinitas veces, obtenemos una sucesión de intervalos cuya intersección es por definición un valor crítico de homología. Llegando a una contradicción.  $\square$

Definimos ahora los *grupos de homología persistente* como  $F_x^y = \text{img } f_x^y$ , es decir, la imagen de  $F_x$  en  $F_y$  por  $f_x^y$ ; surgen de forma natural los *números de Betti persistentes*, definidos como  $\beta_x^y = \dim F_x^y$ . Por convenio,  $F_x^y = \{0\}$  para el caso en que  $x$  o  $y$  son infinito. Consideremos ahora una función “tame”  $f : X \rightarrow \mathbb{R}$ , sean  $(C_i)_{i=1, \dots, n}$  sus valores críticos de homología y sea  $(a_i)_{i=0, \dots, n}$  una sucesión de valores reales tales que  $a_{i-1} < C_i < a_i$  para cada  $i$ . Por conveniencia, definimos  $a_{-1} = C_0 = -\infty$  y  $a_{n+1} = C_{n+1} = +\infty$ . Dados dos enteros  $0 \leq i < j \leq n+1$ , se define la *multiplicidad*  $\mu_i^j$  del par  $(C_i, C_j)$  como

$$\mu_i^j = \beta_{a_{i-1}}^{a_j} - \beta_{a_i}^{a_j} + \beta_{a_i}^{a_{j-1}} - \beta_{a_{i-1}}^{a_{j-1}}.$$

**Definición 1.27.** El *diagrama de persistencia*  $D(f) \subset \overline{\mathbb{R}^2}$  de  $f$  es el multiconjunto de puntos  $(C_i, C_j)$  donde cada punto se cuenta con multiplicidad  $\mu_i^j$ , y los puntos de la diagonal  $\Delta$ , cada uno con multiplicidad infinita. En relación al código de barras, cada barra define un punto  $(b, d)$  del diagrama cuya multiplicidad es el número de barras iguales. Véase que los puntos críticos son los únicos lugares donde pueden nacer o morir barras.

La suma de las multiplicidades de un multiconjunto  $M$  será llamada *multiplicidad total* de  $M$ , denotado por  $\#(M)$ . De esta forma, definimos el *tamaño* de un diagrama de persistencia mediante  $\#(D(f) - \Delta) = \sum_{i < j} \mu_i^j$ .

**Proposición 1.28.** *Sea  $f$  una función “tame”,  $Q_x^y = [-\infty, x] \times [y, \infty]$  y supongamos  $x < y$  que no son valores críticos de homología de  $f$ . Entonces,  $\#(D(f) \cap Q_x^y) = \beta_x^y$ .*

*Demostración.* Sin pérdida de generalidad, podemos asumir que  $x = a_i$  e  $y = a_{j-1}$ . La multiplicidad total que queremos calcular será:

$$\begin{aligned} \#(D(f) \cap Q_x^y) &= \sum_{m \leq i \leq j \leq n} \mu_m^m = \sum_{m \leq i \leq j \leq n} (\beta_{a_{m-1}}^{a_m} - \beta_{a_m}^{a_m} + \beta_{a_m}^{a_{m-1}} - \beta_{a_{m-1}}^{a_{m-1}}) \\ &= \beta_{a_{-1}}^{a_{n+1}} - \beta_{a_i}^{a_{n+1}} + \beta_{a_i}^{a_{j-1}} - \beta_{a_{-1}}^{a_{j-1}} = \beta_{a_i}^{a_{j-1}} = \beta_x^y \end{aligned}$$

donde, analizando cuidadosamente las definiciones, es un simple ejercicio ver como estamos ante una suma telescópica, sobreviviendo solamente los indicados, de los cuales solo uno es no nulo.  $\square$

*Observación 1.29.* Las consecuencias de esta proposición se estudian con más detalle en [16], de donde extraemos la siguiente idea para entender el concepto de multiplicidad. Si la escribimos de la forma  $\mu_i^j = (\beta_{a_i}^{a_j-1} - \beta_{a_i}^{a_j}) - (\beta_{a_{i-1}}^{a_j-1} - \beta_{a_{i-1}}^{a_j})$ . Interpretamos  $\beta_{a_i}^{a_j-1}$  como el número de clases de homología independientes en  $F_{C_{j-1}}$  que nacen antes de  $F_{C_i}$ , si le restamos el número de clases en  $F_{C_j}$  que también nacen antes de  $F_{C_i}$ , nos estamos quedando con el número de clases que nacen antes de  $F_{C_i}$  y mueren en el camino entre  $F_{C_{j-1}}$  y  $F_{C_j}$ . Análogamente,  $(\beta_{a_{i-1}}^{a_j-1} - \beta_{a_{i-1}}^{a_j})$  representará el número de clases que nacen antes de  $F_{C_{i-1}}$  y mueren entre  $F_{C_{j-1}}$  y  $F_{C_j}$ . Luego, la multiplicidad  $\mu_i^j$  será el número de clases de homología independientes que nacen entre  $F_{C_{i-1}}$  y  $F_{C_i}$  y mueren entre  $F_{C_{j-1}}$  y  $F_{C_j}$ .

Nuestro objetivo último es dar un resultado sobre la estabilidad de los diagramas de persistencia, para ello debemos de definir formalmente el concepto de distancia entre dos multiconjuntos de  $\mathbb{R}^2$ . Dados dos puntos  $x = (x_1, x_2)$  e  $y = (y_1, y_2)$  en  $\mathbb{R}^2$ , definimos  $\|x - y\|_\infty = \max\{|x_1 - y_1|, |x_2 - y_2|\}$ . Para dos funciones reales  $f$  y  $g$ , tendremos  $\|f - g\|_\infty = \sup_x |f(x) - g(x)|$ .

**Definición 1.30.** La *distancia Hausdorff* entre dos multiconjuntos  $X, Y \subset \mathbb{R}^2$  es

$$d_H(X, Y) = \max\{\sup_x \inf_y \|x - y\|_\infty, \sup_y \inf_x \|x - y\|_\infty\}.$$

**Definición 1.31.** Sea  $\Gamma$  el conjunto de todas las biyecciones entre dos multiconjuntos  $X$  e  $Y$  de  $\mathbb{R}^2$ , la *distancia bottleneck* entre  $X$  e  $Y$  es

$$d_B(X, Y) = \inf_{\gamma \in \Gamma} \sup_x \|x - \gamma(x)\|_\infty.$$

En ambas definiciones, cada punto con multiplicidad  $\mu$  es considerado como  $\mu$  puntos individuales, teniéndolos en cuenta de esta forma para construir las biyecciones. Observemos que la distancia bottleneck cumple una restricción a mayores respecto de la distancia Hausdorff por calcularse sobre las biyecciones, luego se tiene  $d_H(X, Y) \leq d_B(X, Y)$ . Enunciamos el teorema principal de esta sección, cuya demostración se hará en varios pasos y con ayuda de algunos lemas.

**Teorema 1.32. Estabilidad para la distancia bottleneck.** *Sea  $X$  un espacio triangulable y  $f, g$  dos funciones reales continuas “tame” definidas sobre  $X$ . Entonces los diagramas de persistencia de  $f$  y  $g$  cumplen*

$$d_B(D(f), D(g)) \leq \|f - g\|_\infty.$$

El resultado se puede generalizar para  $X$  un espacio topológico triangulable, como se enuncia en [4]; para ello se tendría que introducir con más detalle la homología singular, aunque daremos una idea de cómo se procedería. Antes de probar el Teorema 1.32, necesitamos algunos lemas.

**Lema 1.33. (del cuadrante) [17]** Sea  $\epsilon = \|f - g\|_\infty$  y dados números reales  $x < y$  que no sean valores críticos de homología, se tiene

$$\#(D(f) \cap Q_{x-\epsilon}^{y+\epsilon}) \leq \#(D(g) \cap Q_x^y).$$

Lo cual viene a decir que la multiplicidad total del diagrama de persistencia de  $g$  en el cuadrante superior izquierdo con esquina en  $(x, y)$  está acotada inferiormente por la multiplicidad total de  $D(f)$  en ese mismo cuadrante encogido por  $\epsilon$ .

Sin embargo, este lema aún no es suficiente para probar la estabilidad con la distancia Hausdorff. En [4] se introduce una modificación del Lema del cuadrante, donde en vez de considerar el cuadrante superior izquierdo y encoger la esquina, se hace para una caja cualquiera de  $\mathbb{R}^2$ , reduciendo todas sus esquinas por  $\epsilon$ .

**Lema 1.34. (de la caja) [17]** Consideremos los siguientes números reales que no sean valores críticos de homología  $a < b < c < d$ . Sea  $B = [a, b] \times [c, d]$  la caja cerrada en el plano extendido y  $B_\epsilon = [a + \epsilon, b - \epsilon] \times [c + \epsilon, d - \epsilon]$  la encogida. Entonces

$$\#(D(f) \cap B_\epsilon) \leq \#(D(g) \cap B).$$

*Demostración.* Interpretaremos la multiplicidad total de un diagrama de persistencia  $D(f)$  intersecado con una caja  $[a, b] \times [c, d]$  como la dimensión del espacio vectorial  $F_{a,b}^{c,d}$  que definimos a continuación. Recordemos que la dimensión del grupo de homología persistencia  $F_b^c$  es la multiplicidad total del cuadrante superior izquierdo con esquina en  $(b, c)$ . Si restringimos  $f_c^d : F_c \rightarrow F_d$  al espacio vectorial  $F_b^c$ , obtenemos una sobreyección  $f_b^{c,d} : F_b^c \rightarrow F_b^c$  para la cual, si denotamos por  $F_b^{c,d}$  su núcleo, tenemos que  $\dim F_b^{c,d} = \dim F_b^c - \dim F_b^d$ . Como  $F_a^c \subseteq F_b^c$ , la aplicación  $f_a^{c,d}$  vendrá dada por la restricción de  $f_b^{c,d}$  al espacio vectorial  $F_a^c$ . Luego definimos el espacio cociente

$$F_{a,b}^{c,d} = F_b^{c,d} / F_a^{c,d},$$

cuya dimensión vendrá dada por

$$\dim F_{a,b}^{c,d} = \dim F_b^{c,d} - \dim F_a^{c,d}.$$

De esta forma podremos expresar las multiplicidades de los diagramas de persistencia de cada función intersecados con sus respectivas cajas como

$$\begin{aligned} \#(D(f) \cap B_\epsilon) &= \dim F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon}, \\ \#(D(g) \cap B) &= \dim G_{a,b}^{c,d}. \end{aligned}$$

Para alcanzar la desigualdad buscada, debemos encontrar una sobreyección definida de un subespacio de  $G_{a,b}^{c,d}$  a  $F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon}$ . Consideremos  $E_b^c = \psi_c^{-1}(F_{b-\epsilon}^{c+\epsilon, d-\epsilon})$ , donde  $\psi_c : G_c \rightarrow F_{c+\epsilon}$  es la

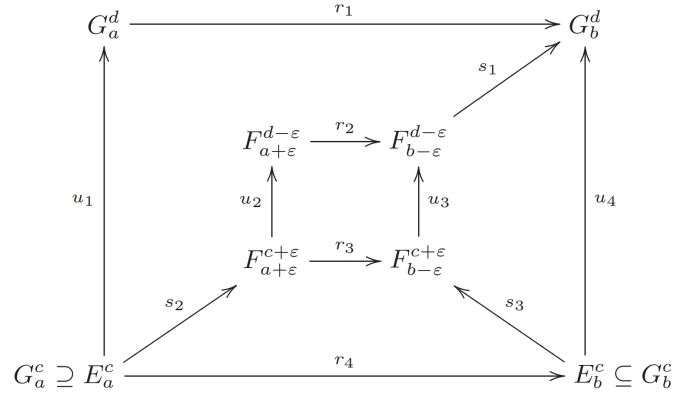


Figura 1.4: Diagrama conmutativo para el Lema de la caja extra (Figura 4 en [4]).

aplicación inducida por la inclusión  $g^{-1}(-\infty, c] \subseteq f^{-1}(-\infty, c + \epsilon]$ . Apoyándonos en el diagrama, sea  $u_3 = f_{b-\epsilon}^{c+\epsilon, d-\epsilon}$ , se tiene que  $E_b^c$  es la preimagen de  $u_3$  por la restricción de  $\psi_c$  a  $G_b^c$ . Ahora, en la prueba del Lema del cuadrante, se obtiene que  $\psi_c(G_b^c) \supseteq F_{b-\epsilon}^{c+\epsilon}$ . Definiendo  $s_3$  como  $\psi_c$  restringida a  $E_b^c$ , se tiene que  $\text{Ker}(u_3) = \text{Img}(s_3)$ .

Sea  $E_a^c = G_a^c \cap E_b^c$ , el cociente  $E_b^c/E_a^c$  es el subespacio buscado sobre el que se define la sobreyección a  $F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon}$ . Volviendo al diagrama, las aplicaciones  $r_i$  denotan la inclusión entre los espacios vectoriales,  $u_2 = f_{a+\epsilon}^{c+\epsilon, d-\epsilon}$ ,  $u_1$  es la restricción de  $g_a^{c,d}$  a  $E_a^c$  y  $u_4$  es la restricción de  $g_b^{c,d}$  a  $E_b^c$ . Otro de los contenidos obtenidos en el Lema del cuadrante es  $\psi_c(G_a^c) \subseteq F_{a+\epsilon}^{c+\epsilon}$  y definimos  $s_2$  como la restricción de  $g_a^{c,d}$  a  $E_a^c$ , luego  $s_2(G_a^c)$  también estará contenido en  $F_{a+\epsilon}^{c+\epsilon}$ . Intercambiando  $G$  por  $F$  en el contenido anterior, se deduce que  $\varphi_{d-\epsilon}(F_{b-\epsilon}^{d-\epsilon}) \subseteq G_b^d$ , donde  $\varphi_{d-\epsilon} : F_{d-\epsilon} \rightarrow G_d$  es la aplicación inducida por la inclusión  $f^{-1}(-\infty, d - \epsilon] \subseteq g^{-1}(-\infty, d)$ ; juntando esto con que  $s_1$  se define como la restricción de  $\varphi_{d-\epsilon}$  a  $F_{b-\epsilon}^{d-\epsilon}$ , se tiene que  $\text{Img}(s_1) \subseteq G_b^d$ .

El diagrama está bien definido y es conmutativo. Se tiene que  $u_4 = s_1 \circ u_3 \circ s_3$ , como  $u_3 \circ s_3 = 0$ , se deduce que  $E_b^c = \text{Ker}(u_4)$ . Por otro camino,  $u_4 \circ r_4 = r_1 \circ u_1$ , como  $u_4 \circ r_4 = 0$  y  $r_1$  es una inclusión, se tiene que  $E_a^c = \text{Ker}(u_1)$ . Entonces,  $E_a^c \subseteq G_a^{c,d}$  y  $E_b^c \subseteq G_b^{c,d}$ . Como por definición  $E_a^c = G_a^c \cap E_b^c$ , se obtiene que  $E_b^c/E_a^c \subseteq G_{a,b}^{c,d}$ . Luego,  $\dim E_b^c/E_a^c \leq \dim G_{a,b}^{c,d}$ .

Como último paso, debemos probar la existencia de la sobreyección buscada. Tengamos en cuenta que  $E_b^c/E_a^c = \text{Ker}(u_4)/\text{Ker}(u_1)$  y  $F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon} = \text{Ker}(u_3)/\text{Ker}(u_2)$ . Por construcción,  $s_3(\text{Ker}(u_4)) = \text{Ker}(u_3)$ , luego si probamos que  $s_3(\text{Ker}(u_1)) \subseteq \text{Ker}(u_2)$  estará claro que  $s_3$  induce una sobreyección. Ahora bien, viendo el diagrama tenemos que  $s_3(\text{Ker}(u_1)) = s_2(\text{Ker}(u_1))$ . Sea  $\nu \in \text{Ker}(u_1)$ , usando la conmutatividad,  $r_2 \circ u_2 \circ s_2(\nu) = u_3 \circ s_3 \circ r_4(\nu) = 0$ . Como  $r_2$  es una inclusión se obtiene el contenido buscado y  $\dim F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon} \leq \dim E_b^c/E_a^c$ .

Finalmente, juntando las desigualdades obtenidas, se concluye la demostración:

$$\dim F_{a+\epsilon, b-\epsilon}^{c+\epsilon, d-\epsilon} \leq \dim E_b^c/E_a^c \leq \dim G_{a,b}^{c,d} \implies \#(D(f) \cap B_\epsilon) \leq \#(D(g) \cap B). \quad \square$$

Presentamos un caso concreto más sencillo del teorema principal en forma de lema, que será usado en su demostración. Sea  $f$  una función real “tame” sobre un espacio topológico  $X$ , consideremos la mínima distancia entre dos puntos diferentes de los que al menos uno está fuera de la diagonal

$$\delta_f = \min\{\|p - q\|_\infty \mid p \in D(f), q \in D(f) - \Delta, p \neq q\}.$$

Diremos que otra función “tame”  $g : X \rightarrow \mathbb{R}$  es *muy cercana* a  $f$  si  $\|f - g\|_\infty < \delta_f/2$ .

**Lema 1.35. (para funciones muy cercanas)** Sean  $f, g : X \rightarrow \mathbb{R}$  con  $g$  muy cercana a  $f$ . Entonces se cumple que  $d_B(D(f), D(g)) \leq \|f - g\|_\infty$ .

*Demostración.* Sea  $\mu$  la multiplicidad de un punto  $p$  en  $D(f) - \Delta$  y  $B_\epsilon$  el cuadrado de centro  $p$  y radio  $\epsilon = \|f - g\|_\infty$ , por el Lema de la caja, tenemos que

$$\mu \leq \#(D(g) \cap B_\epsilon) \leq \#(D(f) \cap B_{2\epsilon}).$$

Pero como  $\epsilon < \delta_f/2$  por la hipótesis de  $g$  muy cercana a  $f$ , tenemos que  $p$  es el único punto en  $D(f) \cap B_{2\epsilon}$  y entonces  $\mu = \#(D(g) \cap B_\epsilon)$ . Luego, para cada punto  $p \in D(f) - \Delta$ , podemos identificar de forma biyectiva todos los puntos de  $D(g) \cap B_\epsilon$  con  $p$  (entendido como  $\mu$  puntos independientes). De esta forma, los únicos puntos de  $D(g)$  sin imagen asociada serán los que disten más de  $\epsilon$  de  $D(f) \cap B_{2\epsilon}$ . Aprovechando que los puntos de la diagonal tienen multiplicidad infinita, llevaremos estos puntos restantes al correspondiente más cercano de  $\Delta$ , que está a distancia menor que  $\epsilon$  por ser  $d_H(D(f), D(g)) \leq \|f - g\|_\infty = \epsilon$ . De este modo, se crea una biyección entre los diagramas de persistencia de  $f$  y  $g$  que mueve cada punto a lo sumo una distancia  $\epsilon$ .  $\square$

Ya casi estamos en condiciones en demostrar la estabilidad para la distancia bottleneck, la idea será construir una biyección como composición de muchas biyecciones como las del Lema 1.35. Antes de ello, probemos un caso particular del enunciado del teorema.

**Lema 1.36. (Caso simplicial)** En las condiciones del Teorema 1.32, sean  $f$  y  $g$  funciones lineales a trozos definidas sobre un complejo simplicial  $T$ , entonces  $d_B(D(f), D(g)) \leq \|f - g\|_\infty$ .

*Demostración.* Definamos la *combinación convexa* de  $f$  y  $g$  como la función  $h_t = (1 - t)f + tg$  para  $t \in [0, 1]$ . La familia de combinaciones convexas para todos los posibles valores de  $t$ , forman una interpolación lineal entre las funciones lineales a trozos iniciales. Queremos descomponer esta interpolación lineal en pasos suficientemente pequeños como para aplicar el Lema para funciones muy cercanas. Denotemos  $\epsilon = \|f - g\|_\infty$  y véase que para cada  $t \in [0, 1]$ ,  $h_t$  es “tame” con  $\delta_{h_t}$  positivo, que escribiremos como  $\delta(t)$ . Consideremos el recubrimiento abierto de  $[0, 1]$  dado por

$$\left\{ U_t = \left( t - \frac{\delta(t)}{4\epsilon}, t + \frac{\delta(t)}{4\epsilon} \right) \right\}.$$

A partir de él, tomemos un subrecubrimiento minimal  $M$ , que será finito por ser  $[0, 1]$  compacto. Establecemos  $t_1 < \dots < t_n$  como los puntos medios de cada intervalo de  $M$ . Por ser  $M$  minimal, para cada par de intervalos consecutivos  $U_{t_i}$  y  $U_{t_{i+1}}$ , su intersección es no vacía. Luego,

$$t_{i+1} - t_i \leq \frac{\delta(t_i) + \delta(t_{i+1})}{4\epsilon} \leq \frac{\max\{\delta(t_i), \delta(t_{i+1})\}}{2\epsilon}.$$

Por definición, tenemos que  $\|h_{t_i} - h_{t_{i+1}}\|_\infty = \epsilon(t_{i+1} - t_i)$ . Lo cual implica por la desigualdad anterior que  $\|h_{t_i} - h_{t_{i+1}}\|_\infty \leq \max\{\delta(t_i), \delta(t_{i+1})\}/2\epsilon$  y por lo tanto  $h_{t_i}$  es muy cercana a  $h_{t_{i+1}}$  o viceversa. Tenemos entonces garantizado por el Lema para funciones muy cercanas que  $d_B(D(h_{t_i}), D(h_{t_{i+1}})) \leq \|h_{t_i} - h_{t_{i+1}}\|_\infty$  para  $1 \leq i \leq n-1$ . El resultado se extiende para  $i \in \{0, n\}$  definiendo  $t_0 = 0$  y  $t_{n+1} = 1$  por ser  $h_0$  muy cercana a  $h_{t_1}$  y  $h_1$  muy cercana a  $h_{t_n}$ . Finalmente aplicando la desigualdad triangular, llegamos a que

$$d_B(D(f), D(g)) \leq \sum_{i=0}^n d_B(D(h_{t_i}), D(h_{t_{i+1}})) \leq \sum_{i=0}^n \|h_{t_i} - h_{t_{i+1}}\|_\infty = \|f - g\|_\infty,$$

concluyendo la demostración.  $\square$

**Demostración del Teorema 1.32.** Por ser  $X$  triangulable, existe un complejo simplicial  $T$  y un homeomorfismo  $\varphi : T \rightarrow X$ . Por hipótesis, tenemos  $f, g$  dos funciones reales continuas “tame” sobre  $X$ . Claramente,  $f \circ \varphi$  también será “tame” y  $D(f) = D(f \circ \varphi)$ . Puesto que  $f$  y  $g$  son continuas y  $T$  es compacto, para  $\delta > 0$  suficientemente pequeño existe una subdivisión  $L$  de  $T$  tal que

$$\begin{aligned} |f \circ \varphi(v) - f \circ \varphi(w)| &\leq \delta, \\ |g \circ \varphi(v) - g \circ \varphi(w)| &\leq \delta, \end{aligned}$$

para  $v, w$  vértices de un mismo símplice en  $L$ . Consideremos ahora  $f^*, g^* : L \rightarrow \mathbb{R}$  las interpolaciones lineales a trozos de  $f \circ \varphi$  y  $g \circ \varphi$  en  $L$ . Que cumplen por construcción  $f^* - f \circ \varphi \leq \delta$  y  $g^* - g \circ \varphi \leq \delta$ . Usando la desigualdad triangular acotamos la distancia bottleneck entre los diagramas de persistencia de  $f$  y  $g$  como

$$d_B(D(f), D(g)) \leq d_B(D(f), D(f^*)) + d_B(D(f^*), D(g^*)) + d_B(D(g^*), D(g)).$$

Los sumandos de los extremos a la derecha de la desigualdad, asumiendo  $\delta < \min\{\delta_f/2, \delta_g/2\}$ , podemos aplicar el Lema para funciones muy cercanas y obtener

$$\begin{aligned} d_B(D(f), D(f^*)) &= d_B(D(f \circ \varphi), D(f^*)) \leq \delta, \\ d_B(D(g^*), D(g)) &= d_B(D(g), D(g^*)) = d_B(D(g \circ \varphi), D(g^*)) \leq \delta. \end{aligned}$$

Para el sumando del medio, ya que estamos en el caso simplicial, se obtiene directamente la acotación  $d_B(D(f^*), D(g^*)) \leq \|f^* - g^*\|_\infty \leq \|f - g\|_\infty + 2\delta$ . Juntando todos los resultados, tenemos

$$d_B(D(f), D(g)) \leq \|f - g\|_\infty + 4\delta.$$

Ahora bien, como podemos hacer  $\delta$  tan pequeño como queramos, se obtiene el resultado deseado.  $\square$

## 1.5. Landscapes

Supongamos que partimos de una variedad topológica  $X$  en un espacio métrico, hemos aprendido que mediante la homología persistente podemos hacer un estudio general de la evolución temporal de los generadores de los grupos de homología de los conjuntos de subnivel dados por una función bajo ciertas condiciones. En concreto, la información resultante consiste en pares de puntos  $(b, d)$  que recogen el tiempo en el que nacen y mueren cada clase de homología independiente. Por ahora, hemos visto que este multiconjunto de pares se puede representar mediante códigos de barras o con los diagramas de persistencia  $D$ , representando los  $(b, d)$  sobre el plano extendido con una multiplicidad concreta.

Existe otra representación que presenta una correspondencia unívoca con los diagramas de persistencia, que son los llamados *landscapes*, que denotaremos por  $\mathcal{L}$ . Luego también será un invariante útil para diferenciar espacios no homeomorfos. Consideremos la siguiente aplicación  $f$  que transforma los puntos de un diagrama persistente mediante

$$f(b, d) = \left( \frac{b+d}{2}, \frac{b-d}{2} \right),$$

donde para cada punto, la coordenada  $x$  representa el tiempo en el que la clase alcanza la mitad de su vida, y la coordenada  $y$  es su semivida. De este modo, el eje  $X$  informa sobre cuándo viven las clases y el eje  $Y$  nos da una idea de cuánto viven.

Ahora, definimos el conjunto  $L$ , donde cada punto  $p = (x, y)$  de la imagen del diagrama de persistencia por  $f$ , se reemplaza por la función triangular

$$T_p(z) = \begin{cases} z - x + y & \text{si } z \in [x - y, x] \\ x + y - z & \text{si } z \in (x, x + y] \\ 0 & \text{en otro caso} \end{cases}$$

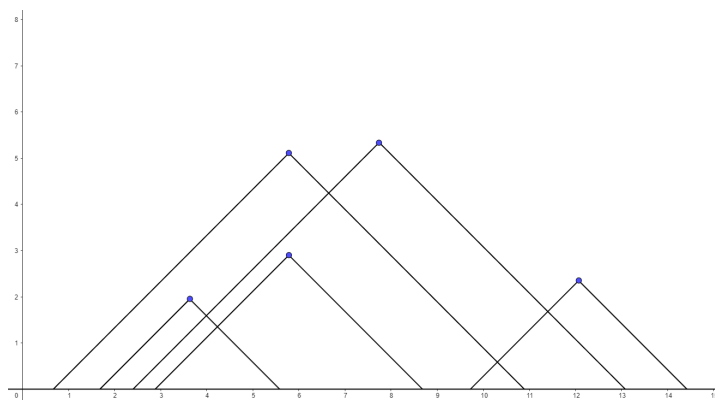


Figura 1.5: Imagen por  $T_p$  del conjunto de puntos  $p = \left( \frac{b+d}{2}, \frac{b-d}{2} \right)$  definido a partir de un diagrama de persistencia  $D$ .

El resultado es el conjunto representado en la figura 1.5, donde además se marcaron en azul los puntos iniciales de la forma  $p = (x, y) = (\frac{b+d}{2}, \frac{b-d}{2})$ . Será a partir de este conjunto que definiremos los landscapes como se indica a continuación.

**Definición 1.37.** El  $k$ -landscape asociado a un diagrama de persistencia  $D$ , denotado por  $\mathcal{L}_D(k, z)$  será la función dada por

$$\mathcal{L}_D(k, z) = k - \max_{q \in D} T_{f(q)}(z),$$

donde  $k - \text{máx}$  es el  $k$ -ésimo máximo en el conjunto.

Véase que se trata de una función 1-Lipschitz. Para entender gráficamente la construcción de estas funciones se presenta la figura 1.6, donde se puede entender fácilmente el procedimiento a seguir. El resultado siempre son esas formas que parecen siluetas de montañas, de ahí el nombre de landscapes, traducido del inglés como paisajes.

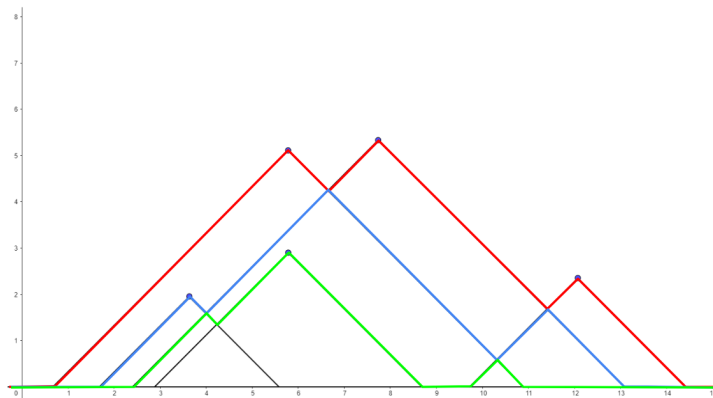


Figura 1.6: Para el mismo conjunto de la figura anterior, se representa  $\mathcal{L}_D(1, z)$  en rojo,  $\mathcal{L}_D(2, z)$  en azul y  $\mathcal{L}_D(3, z)$  en verde. Véase que en ciertos intervalos del eje  $X$  se superponen los colores.

El resultado de estabilidad también se tiene para los landscapes, como se muestra en el Teorema 12 de la Referencia [5]. Dados dos diagramas de persistencia  $D(f)$  y  $D(g)$ , con sus respectivos landscapes  $\mathcal{L}(f)$  y  $\mathcal{L}(g)$ , se tiene que

$$\|\mathcal{L}(f) - \mathcal{L}(g)\|_\infty \leq \|f - g\|_\infty.$$

Además, se muestra que la norma infinito entre dos landscapes está acotada por la distancia bottleneck entre los correspondientes diagramas.

Finalmente, nos interesa la aplicación de los resultados de estabilidad para nubes de puntos, que será con lo que trabajemos experimentalmente. La estabilidad de la filtración de Rips sobre nubes de puntos en espacios métricos fue establecida en [18] pero la versión presentada en ese artículo necesita la introducción de conceptos como distancia de intercalamiento o distancia

Gromov-Hausdorff que van más allá de los intereses de este trabajo. Seguidamente damos una versión más restringida de estabilidad para filtraciones de Rips que será suficiente para el objetivo del trabajo.

Sea  $\mathbb{X}$  un conjunto finito de un espacio métrico  $(X, d)$ . Sea  $T(\mathbb{X})$  el complejo simplicial total sobre  $\mathbb{X}$ , i.e., el complejo simplicial que tiene todos los símlices sobre el conjunto de vértices indexado por  $\mathbb{X}$ .

Sea  $d_{\mathbb{X}} : X \rightarrow \mathbb{R}$  la función (continua) definida como  $d_{\mathbb{X}}(x) = \min_{z \in \mathbb{X}} \{d(x, z)\}$ . Véase que si  $\mathbb{X}$  y  $\mathbb{Y}$  son conjuntos finitos en  $X$  entonces  $\|d_{\mathbb{X}} - d_{\mathbb{Y}}\|_{\infty} = d_H(\mathbb{X}, \mathbb{Y})$ .

Identifiquemos  $\mathbb{X}$  con el conjunto de vértices de  $T(\mathbb{X})$ .

*Afirmación 1.38.* Existe una extensión  $\bar{d}_{\mathbb{X}} : T(\mathbb{X}) \rightarrow \mathbb{R}$  y una sucesión creciente de números reales  $t_n$  tal que: (1)  $(\bar{d}_{\mathbb{X}})^{-1}((-\infty, t_n]) = VR(\mathbb{X}, t_n)$ ; (2) si  $t \in [t_n, t_{n+1})$  entonces  $(\bar{d}_{\mathbb{X}})^{-1}((-\infty, t])$  es homotópico a  $(\bar{d}_{\mathbb{X}})^{-1}((-\infty, t_n])$ .

Observemos ahora que la construcción de Vietoris Rips también funciona para pseudo-métricas. La diferencia entre una pseudo-métrica y una métrica es que pueden existir puntos distintos a distancia nula.

Sean  $\mathbb{X}$  y  $\mathbb{Y}$  dos subconjuntos finitos de  $(X, d)$ . Sea  $\epsilon = d_H(\mathbb{X}, \mathbb{Y})$ . Sea  $\iota : \mathbb{Y} \rightarrow \mathbb{X}$  cualquier aplicación que cumpla  $d(y, \iota(y)) < \epsilon$ . Definamos una pseudométrica en  $\mathbb{X} \cup \mathbb{Y}$  como

$$d_{\iota}(x, y) = \begin{cases} d(x, y) & \text{si } x, y \in \mathbb{X} \\ d(x, \iota(y)) & \text{si } x \in \mathbb{X} \text{ } y \in \mathbb{Y} \\ d(\iota(x), \iota(y)) & \text{si } x, y \in \mathbb{Y} \end{cases}$$

*Afirmación 1.39.* Existe una extensión  $\bar{d}_{\iota} : T(\mathbb{X} \cup \mathbb{Y}) \rightarrow \mathbb{R}$  tal que  $(\bar{d}_{\mathbb{X}})^{-1}((-\infty, t])$  es homotópico a  $\bar{d}_{\iota}^{-1}((-\infty, t])$  para todo  $t \in \mathbb{R}$  y  $\|\bar{d}_{\iota} - \bar{d}_{\mathbb{X} \cup \mathbb{Y}}\|_{\infty} = d_H(\mathbb{X}, \mathbb{Y})$ .

Combinando los resultados anteriores junto con el Teorema de estabilidad presentado obtenemos que:

$$\begin{aligned} d_B(VR(\mathbb{X}_d), VR(\mathbb{Y}_d)) &= d_B(VR(\mathbb{X} \cup \mathbb{Y}_{d_{\iota}}), VR(\mathbb{X} \cup \mathbb{Y}_{d_j})) \leq \|\bar{d}_{\iota} - \bar{d}_j\|_{\infty} \leq \\ &\leq \|\bar{d}_{\iota} - \bar{d}_{\mathbb{X} \cup \mathbb{Y}}\|_{\infty} + \|\bar{d}_{\mathbb{X} \cup \mathbb{Y}} - \bar{d}_j\|_{\infty} = 2d_H(\mathbb{X}, \mathbb{Y}). \end{aligned}$$

Siendo  $d_B$  la distancia Bottleneck entre diagramas de persistencia,  $j : \mathbb{X} \rightarrow \mathbb{Y}$  cualquier aplicación que cumpla  $d(x, j(x)) < \epsilon$ ,  $d_j$  la pseudométrica en  $\mathbb{X} \cup \mathbb{Y}$  definida de forma análoga a  $d_{\iota}$  y  $VR(Z_d)$  denota la construcción de Rips sobre el conjunto  $Z$  para una pseudométrica  $d$ .

Por el mismo razonamiento obtenemos estabilidad para landscapes:

**Proposición 1.40.**  $\|\mathcal{L}(VR(\mathbb{X})) - \mathcal{L}(VR(\mathbb{Y}))\|_{\infty} \leq 2d_H(\mathbb{X}, \mathbb{Y})$ .

## 1.6. Tratamiento estadístico

Queremos usar la homología persistente para estudiar la topología y métrica, tanto global como local, de una variedad Riemanniana compacta  $X$ . Para ello, una opción es considerar el landscape  $\mathcal{L}$  de una muestra aleatoria uniforme (donde la densidad uniforme viene dada por la forma de volumen de la variedad) de un número fijo de puntos sobre  $X$  asociado a una filtración;  $\mathcal{L}$  será una variable aleatoria perteneciente a una distribución de probabilidad a priori desconocida. Podemos definir el landscape promedio  $\mu_X$ , entendido como la esperanza de la variable aleatoria. Para estimarlo, lo que haremos será generar  $N$  muestras aleatorias de puntos sobre la variedad, y para cada una de ellas calcular el diagrama persistente asociado, generando así una muestra aleatoria de tamaño  $N$  sobre la distribución de probabilidad.

Sea  $Y$  otra variedad Riemanniana compacta, observar que si  $f : X \rightarrow Y$  es una isometría entonces el pullback de  $L_Y$  es igual a  $L_X$ . Se sigue que  $\mu_X = \mu_Y$ , por tanto es un invariante métrico. Como obtenerlo analíticamente puede ser complicado, lo estimaremos experimentalmente. En general, este objeto no es un landscape, ya que es una función que no tiene por qué estar asociada a un diagrama. Por otra parte, el diagrama persistente promedio no es fácil de definir, razón principal por la que nos decantamos por los landscapes, que son simples funciones.

Siguiendo [6], el objetivo en esta sección es dar bandas de confianza para los landscapes promedios a partir de muestras, que nos garanticen para un nivel de significación  $\alpha$ , que la probabilidad de que  $\mu$  se encuentre en esa región es de  $1 - \alpha$ .

El bootstrap estándar es una herramienta estadística para calcular intervalos de confianza, introducida en [19]. Supongamos las variables aleatorias independientes e idénticamente distribuidas  $X_1, \dots, X_N$  que toman valores sobre un espacio de medida  $(\mathbb{X}, \mathcal{X}, P)$ . Si queremos estimar un parámetro real  $\theta$  de la distribución  $P$ , usaríamos el estadístico  $\hat{\theta} = h(X_1, \dots, X_N)$ , usando los datos conocidos. Conocer la distribución de  $\hat{\theta} - \theta$  será suficiente para construir los intervalos de confianza, pero esta depende de la distribución  $P$ , que será desconocida.

Ahora bien, a partir de los datos empíricos  $X_1, \dots, X_N$  podemos aproximar  $P$  mediante la distribución empírica  $P_N$  que otorga probabilidad uniforme  $1/N$  a cada  $X_i$  de la muestra. De este modo, el bootstrap estándar para dar intervalos de confianza sobre un parámetro  $\theta$  a partir de una muestra  $X_1, \dots, X_N$  consiste en los siguientes pasos:

- Calcular el estimador  $\hat{\theta} = h(X_1, \dots, X_N)$ .
- Estimar  $P$  mediante  $P_N$  y sacar de ella una nueva muestra  $X_1^*, \dots, X_N^*$ .
- Calcular  $\hat{\theta}_1^* = h(X_1^*, \dots, X_N^*)$  y repetir el proceso  $B$  veces para obtener  $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$ .
- Idealmente, estimaríamos la función de distribución acumulada  $F$  de  $\hat{\theta} - \theta$ , mediante el estadístico  $\hat{F}(q) = P_N(\hat{\theta}^* - \theta \leq q)$ , pero no es fácilmente calculable analíticamente. Aproximi-

maremos este valor por  $\overline{F}_B(q) = \frac{1}{B} \sum_{k=1}^B I(\hat{\theta}_k^* - \hat{\theta} \leq q)$ , siendo  $I$  la función característica.

- Construir el intervalo de confianza  $C_N = \left[ \hat{\theta} - \overline{F}_B^{-1}(1 - \alpha/2), \hat{\theta} - \overline{F}_B^{-1}(\alpha/2) \right]$  a nivel  $1 - \alpha$ .

En el proceso hemos realizado dos estimaciones, la de  $F$  por  $\hat{F}$  y la de  $\hat{F}$  por  $\overline{F}$ . La segunda no es un problema como fuente de error, puesto que para reducirlo simplemente tenemos que tomar  $B$  suficientemente grande. Para demostrar estadísticamente la validez del método, se debe probar que  $\sup_q |\overline{F}(q) - F(q)|$  converge en probabilidad a 0, y por lo tanto,  $\liminf_{N \rightarrow \infty} \mathbb{P}(\theta \in C_N) \geq 1 - \alpha$ , donde  $\mathbb{P}$  denota la probabilidad de un suceso.

El bootstrap estándar nos proporciona un método para dar un intervalo de confianza en el caso de que nuestra variable aleatoria sea una función, que consiste en aplicarlo sobre cada punto del dominio. Sin embargo, este debe ser adaptado para tratar con funciones, siendo válido bajo las condiciones formalmente explicadas en [6]. Sea  $f$  una función real sobre el espacio de medida  $(\mathbb{X}, \mathcal{X}, P)$ , usando la notación  $Pf = \int f dP$  y  $P_N f = \frac{1}{N} \sum_{i=1}^N f(X_i)$  donde  $X_1, \dots, X_N$  son variables aleatorias independientes e idénticamente distribuidas, y definamos  $\mathbb{G}_N f = \sqrt{N}(P_N f - Pf)$ , al cual llamaremos proceso empírico. El procedimiento empírico para construir una banda de confianza a nivel  $1 - \alpha$  para  $Pf$  es el siguiente (bajo ciertas condiciones de  $f$  que garantizan la convergencia del proceso empírico, expuestas en el Teorema 1.4 de [6]):

- Usar la estimación  $P_N$  a partir de la muestra  $X_1, \dots, X_N$  de la misma forma que en el bootstrap estándar.
- A partir de  $P_N$ , obtener una nueva muestra  $X_1^*, \dots, X_N^*$ , para calcular  $\hat{\Theta}_1^* = \sup_f |\mathbb{G}_N^* f|$ , donde  $\mathbb{G}_N^* f = \sqrt{N}(P_N^* f - P_N f)$ .
- Repetir estos pasos  $B$  veces para obtener  $\hat{\Theta}_1^*, \dots, \hat{\Theta}_B^*$ .
- Calcular el cuantil  $q_\alpha = \inf \left\{ q \mid \frac{1}{B} \sum_{k=1}^B I(\hat{\Theta}_k^* \geq q) \leq \alpha \right\}$ .
- La banda de confianza será  $C_N(f) = \left[ P_N f - \frac{q_\alpha}{\sqrt{N}}, P_N f + \frac{q_\alpha}{\sqrt{N}} \right]$ .

Véase que de forma similar al bootstrap estándar, hemos usado  $\hat{\Theta}^*$  para aproximar de forma empírica el estadístico  $\hat{\Theta} = \sup_f |\mathbb{G}_N f|$ . En particular, esta técnica es válida para dar regiones de confianza sobre landscapes, que será utilizado posteriormente en la parte experimental. Supongamos que trabajamos sobre un  $k$ -landscape con  $k$  fijo, sean  $\mathcal{L}_1, \dots, \mathcal{L}_N$  una muestra, asociada a otra  $D_1, \dots, D_N$  de diagramas de persistencia a partir de una distribución  $L$  sobre el espacio de diagramas de persistencia. Definiendo el *landscape promedio empírico* como  $\overline{\mathcal{L}}_N = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i$  y tomando  $\hat{\Theta}^* = \sup_{t \in \mathbb{R}} |\sqrt{N}(\overline{\mathcal{L}}_N^*(t) - \overline{\mathcal{L}}_N(t))|$ , la banda de confianza a nivel  $1 - \alpha$  para la media  $\mu = \mathbb{E}_L[\mathcal{L}_i]$  será

$$C_N = \left[ \overline{\mathcal{L}}_N - \frac{q_\alpha}{\sqrt{N}}, \overline{\mathcal{L}}_N + \frac{q_\alpha}{\sqrt{N}} \right],$$

siguiendo la notación y los pasos anteriores.

## Capítulo 2

# Geometría hiperbólica

En este capítulo cambiamos el tema drásticamente hacia la geometría hiperbólica. En primer lugar, definiremos las variedades hiperbólicas y caracterizaremos su cubierta universal bajo los modelos más comunes, explicando los conceptos básicos.

Después nos centraremos en la demostración del teorema principal del trabajo: el Teorema de Mostow, el cual establece que dos variedades hiperbólicas compactas de dimensión  $n \geq 3$  homotópicamente equivalentes son isométricas. Las implicaciones de este teorema son la base para mezclar con éxito en la parte experimental los campos de la homología persistente y la geometría hiperbólica. Antes de la propia demostración, será necesario presentar algunas definiciones y resultados que mencionaremos durante la misma.

### 2.1. Variedades hiperbólicas y cubierta universal

Una variedad hiperbólica es una variedad Riemanniana geodésicamente completa y con curvatura constante  $-1$ . Esta última propiedad, dependiendo de la literatura consultada, no es parte de la definición y se debe especificar.

Salvo isometrías, existe una única variedad hiperbólica de dimensión  $n$  simplemente conexa, que llamaremos espacio hiperbólico de dimensión  $n$ . Por lo tanto, este será la cubierta universal de cualquiera  $n$ -variedad hiperbólica. A continuación se presentan las principales construcciones de esta cubierta universal:

**El modelo del disco de Poincaré  $\mathbb{D}^n$ :** Consiste en considerar el siguiente subespacio de  $\mathbb{R}^n$

$$\{x \in \mathbb{R}^n \mid \|x\| < 1\}$$

dotado de la métrica hiperbólica mediante el tensor métrico de Riemman:

$$ds^2 = 4 \frac{\sum_i dx_i^2}{(1 - \sum_i x_i^2)^2}$$

Definiremos el borde del infinito  $\partial\mathbb{D}^n = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$ , llamada también esfera del infinito  $S_\infty^{n-1}$  y sus puntos son los denominados puntos ideales.

Bajo este modelo, las líneas rectas para la geometría euclidiana consisten o bien en diámetros del disco o bien curvas de circunferencia ortogonales. De este modo, si consideramos el disco y su borde, una geodésica completa  $\gamma$  queda determinada por sus finales, entendidos como los límites en el borde del disco. Así, dados dos puntos ideales, existirá una única geodésica que los una.

**El modelo del semiespacio  $\mathbb{H}^n$ :** Consiste en el semiespacio superior definido por el conjunto

$$\{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_n > 0\}$$

bajo la métrica:

$$ds^2 = \frac{\sum_i dx_i^2}{x_n^2}$$

Para este modelo, se define el borde del infinito como  $\{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_n = 0\} \cup \infty$ . Análogamente, las líneas rectas serán curvas de circunferencia o líneas verticales, ambas ortogonales al hiperplano  $x_n = 0$ . Las geodésicas completas serán del primer tipo cuando sus finales sean dos puntos distintos al infinito, y del segundo cuando uno de los puntos ideales que la determinan sea el infinito.

Sea  $p \in \mathbb{R}^n \mid \forall i \neq n \ x_i = 0, x_n = -1$ , la isometría entre ambas construcciones es la siguiente:

$$\begin{aligned} \mathbb{D}^n &\rightarrow \mathbb{H}^n \\ x &\mapsto p + \frac{2(x-p)}{\|x-p\|^2} \end{aligned}$$

**El modelo del disco de Klein:** Este modelo combina propiedades de los dos anteriores. Los puntos pertenecen al disco unidad y las líneas rectas hiperbólicas son también las líneas rectas desde el punto de vista euclídeo. No se trata de un modelo conforme ya que no preserva los ángulos, al contrario de las construcciones anteriores, por ello, no será utilizado en este capítulo para las demostraciones.

Sin embargo, en la parte experimental, será necesario conocer la transformación de los puntos de este modelo al del disco de Poincaré  $\mathbb{D}^n$ , que es la siguiente: sea  $x$  un punto del disco bajo el modelo de Klein y  $|x|$  su distancia euclídea respecto al centro del disco, entonces las coordenadas de  $x$  en el modelo del disco de Poincaré serán las del punto situado en la misma dirección radial que  $x$  pero a una distancia del centro

$$r = \frac{|x|}{1 + \sqrt{1 - |x|^2}}$$

Véase, que los puntos ideales se mantienen en sí mismos bajo el cambio de un modelo al otro.

En la práctica trabajaremos con variedades hiperbólicas de dimensión 3. Estudiemos concretamente cómo son las isometrías en el espacio hiperbólico de  $n = 3$ . Consideramos el modelo del semiespacio  $\mathbb{H}^3$ , en [21] se explica que toda isometría que preserve la orientación viene dada por elementos de  $PSL(2, \mathbb{C})$ , que son matrices  $2 \times 2$  con determinante 1 cocientados por  $\pm I$ .

Una transformación de Möbius se define sobre el plano complejo extendido  $\hat{\mathbb{C}} = \mathbb{C} \cup \infty$  como una función tal que

$$z \mapsto \frac{az + b}{cz + d}$$

con  $a, b, c, d, z \in \mathbb{C}$  y  $ad - cd \neq 0$ , y forman precisamente el grupo  $PSL(2, \mathbb{C})$ . Considerando el borde del infinito  $\partial\mathbb{H}^3$  como  $\hat{\mathbb{C}}$ , se tiene que toda isometría de  $\mathbb{H}^3$  se puede ver como la extensión de una aplicación conforme de  $\hat{\mathbb{C}}$  en sí mismo, ya que debe llevar semiesferas con ecuador en  $\hat{\mathbb{C}}$  a semiesferas con ecuador en  $\hat{\mathbb{C}}$ , luego lleva circunferencias en  $\hat{\mathbb{C}}$  a circunferencias en  $\hat{\mathbb{C}}$ . Para definir la extensión a  $\mathbb{H}^3$ , introducimos las siguientes funciones de Möbius simples generadoras de  $\hat{\mathbb{C}}$  en  $\hat{\mathbb{C}}$ :

$$(i) \quad z \rightarrow z + \lambda \quad (ii) \quad z \rightarrow \lambda z \quad (iii) \quad z \rightarrow -\frac{1}{z}$$

donde  $\lambda \in \mathbb{C}$ . Y sus respectivas extensiones a  $\mathbb{H}^3$ , que llamaremos isometrías generadoras, son:

$$(i) \quad (z, x_3) \rightarrow (z + \lambda, x_3) \quad (ii) \quad (z, x_3) \rightarrow (\lambda z, |\lambda|x_3) \quad (iii) \quad (z, x_3) \rightarrow \left( -\frac{\bar{z}}{|z|^2 + x_3^2}, \frac{x_3}{|z|^2 + x_3^2} \right)$$

donde  $z = x_1 + ix_2$ . Vemos que toda transformación de Möbius se puede expresar como composición de las simples. Sea  $f(z) = \frac{az+b}{cz+d}$ , consideremos  $f_1, f_2, f_3, f_4$  como:

$$f_1(z) = z + \frac{d}{c} \quad f_2(z) = -\frac{1}{z} \quad f_3(z) = -\left(\frac{b}{c} - \frac{ad}{c^2}\right) \cdot z \quad f_4(z) = z + \frac{a}{c}$$

entonces  $f = f_4 \circ f_3 \circ f_2 \circ f_1$ . Haciendo las cuentas:

$$\begin{aligned} f_4(f_3(f_2(f_1(z)))) &= f_4\left(f_3\left(f_2\left(z + \frac{d}{c}\right)\right)\right) \\ &= f_4\left(f_3\left(-\frac{1}{z + \frac{d}{c}}\right)\right) = f_4\left(f_3\left(-\frac{c}{cz + d}\right)\right) \\ &= f_4\left(\frac{c\left(\frac{b}{c} - \frac{ad}{c^2}\right)}{cz + d}\right) = f_4\left(\frac{b - \frac{ad}{c}}{cz + d}\right) \\ &= \frac{b - \frac{ad}{c}}{cz + d} + \frac{a}{c} = \frac{b - \frac{ad}{c} + az + \frac{ad}{c}}{cz + d} = \frac{az + b}{cz + d} \end{aligned}$$

Luego tendremos definida la extensión única a una isometría de cualquier transformación de Möbius. De este modo se obtiene la correspondencia buscada, concluyendo que el grupo de isometrías de  $\mathbb{H}^3$  que preservan la orientación es  $PSL(2, \mathbb{C})$ . Definimos los siguientes tipos de isometrías, sea  $\alpha \in PSL(2, \mathbb{C})$ , puede ser:

- Parabólica  $\iff \alpha$  tiene un único punto fijo en  $\hat{\mathbb{C}}$ .
- Hiperbólica  $\iff \alpha$  tiene dos puntos fijos en  $\hat{\mathbb{C}}$  pero ninguno en  $\mathbb{H}^3$ .
- Elíptica  $\iff \alpha$  deja fija puntualmente una geodésica de  $\mathbb{H}^3$ .

Finalmente, presentamos el Teorema de Liouville junto a un corolario que será útil posteriormente durante la demostración del Teorema de Mostow. Dado  $p \in \mathbb{R}^n$  y  $r > 0$ , se define la inversión con centro en  $p$  y radio  $r$  como:

$$i_{p,r} : \mathbb{R}^n - \{p\} \rightarrow \mathbb{R}^n - \{p\}$$

$$x \mapsto p + \frac{r^2(x-p)}{\|x-p\|^2}$$

Véase que una inversión es una aplicación conforme.

**Teorema 2.1.** (Liouville) *Para dimensión  $n \geq 3$ , todo difeomorfismo conforme entre dos subespacios abiertos y conexos de  $\mathbb{R}^n$  es de la forma*

$$x \mapsto \lambda Ai(x) + b$$

con  $\lambda > 0$ ,  $A \in O(n)$ ,  $i$  una inversión o la identidad.

$\mathbb{D}^n$  y  $\mathbb{H}^n$  pueden ser considerados como subvariedades abiertas de  $\mathbb{R}^n$  mediante un embebiendo conforme. Luego las isometrías de  $\mathbb{D}^n$  y  $\mathbb{H}^n$  son de la forma del Teorema de Liouville y se pueden extender a un homeomorfismo conforme de  $\overline{\mathbb{R}^n}$  en sí mismo. Luego toda isometría tiene asociada un homeomorfismo conforme en el borde.

Aplicando el Teorema de Liouville, todo homeomorfismo conforme de  $\overline{\mathbb{R}^n}$  en sí mismo será de la forma  $x \mapsto \lambda Ai(x) + b$ , luego todo homeomorfismo conforme de  $\mathbb{D}^n$  o  $\mathbb{H}^n$  en sí mismo se puede extender a un homeomorfismo conforme de  $\mathbb{D}^n$  o  $\mathbb{H}^n$  en sí mismo, respectivamente. Se puede probar que esta extensión es una isometría hiperbólica. El resultado que usaremos posteriormente en la demostración del Teorema de Mostow se recoge en el siguiente corolario.

**Corolario 2.2.** *Para  $n \geq 3$ , hay una correspondencia uno a uno entre los homeomorfismos conformes de  $\partial\mathbb{D}^n$  y  $\partial\mathbb{H}^n$  y las isometrías de las variedades hiperbólicas completas y simplemente conexas.*

## 2.2. Aplicaciones cuasiconformes

En esta sección introducimos algunos resultados básicos sobre aplicaciones cuasiconformes siguiendo [23], que serán utilizados en la demostración del Teorema de Mostow. El objetivo no es profundizar sobre este tema, para ello se puede consultar la referencia [25], por lo que solo se enuncian los resultados sin demostrar.

**Definición 2.3.** Sean  $\Omega_1, \Omega_2$  dos subconjuntos abiertos de  $\mathbb{R}^n$ . Dado  $f : \Omega_1 \rightarrow \Omega_2$  un homeomorfismo. Definimos para  $x \in \Omega_1$ :

$$\begin{aligned} L(x, y) &= \limsup_{y \rightarrow x} \frac{|f(y) - f(x)|}{|y - x|} \\ l(x, y) &= \liminf_{y \rightarrow x} \frac{|f(y) - f(x)|}{|y - x|} \\ H(x, f) &= \limsup_{r \rightarrow 0} \frac{\sup_{|y-x|=r} |f(y) - f(x)|}{\inf_{|y-x|=r} |f(y) - f(x)|} \end{aligned}$$

Si además  $f$  es diferenciable en  $x$ , denotaremos por  $J(x, f)$  al determinante de la matriz jacobiana de  $f$  en  $x$ .

**Definición 2.4.** Sea  $\Omega$  subconjunto abierto de  $\mathbb{R}^n$  y  $f$  una función real continua definida en  $\Omega$ . Diremos que  $f$  es *Absolutamente Continua en Líneas (ACL)* si  $f$  es absolutamente continua sobre casi todo segmento de línea paralelo a un eje de coordenadas.

Toda función ACL tiene derivadas parciales en casi todo punto, las cuales son localmente integrables. Se tiene el siguiente teorema que nos da en cierto modo la implicación en el otro sentido:

**Teorema 2.5.** Si para una función continua  $f$ , sus derivadas parciales distribucionales son localmente integrales, entonces  $f$  es ACL.

Para ver la demostración, consultar [26]. En ella se hace el caso  $n = 2$ , que se puede generalizar para cualquier  $n$ .

Extendiendo la definición anterior, diremos que una aplicación  $f : \Omega \rightarrow \mathbb{R}^m$  es ACL si es ACL para cada coordenada de  $\mathbb{R}^m$ . Se tiene que la propiedad de ser ACL es invariante bajo cambios de coordenadas, luego se puede extender esta propiedad a aplicaciones entre variedades.

**Definición 2.6.** Sea  $f : \Omega_1 \rightarrow \Omega_2$  un homeomorfismo entre subconjuntos abiertos de  $\mathbb{R}^n$ . Diremos que  $f$  es *cuasiconforme* si  $\exists K > 0$  tal que  $f$  es ACL, diferenciable en casi todo punto y para casi todo punto  $x \in \Omega_1$  se tiene que

$$\frac{L(x, f)^n}{K} \leq |J(x, f)| \leq K \cdot l(x, f)^n,$$

el término  $K$  – *cuasiconforme* se usa para indicar la  $K$  cumpliendo lo anterior.

En el siguiente teorema, se presenta una caracterización que utilizaremos posteriormente para no tener que probar por definición la cuasiconformidad de una aplicación.

**Teorema 2.7.** (Demostrado en [25]) *Sea  $f : \Omega_1 \rightarrow \Omega_2$  un homeomorfismo entre dos conjuntos abiertos de  $\mathbb{R}^n$ . Las siguientes afirmaciones son equivalentes:*

- (1)  $H(x, f)$  está acotado  $\forall x \in \Omega_1$ .
- (2)  $\exists K > 0$ , tal que  $f$  es  $K$ -cuasiconforme.

Otro resultado que será de gran utilidad durante la demostración del Teorema de Mostow, se recoge a continuación.

**Teorema 2.8.** (Demostrado en [27] Págs. 101-102) *Toda aplicación 1-cuasiconforme de  $S^n$  en  $S^n$  es conforme.*

Como consecuencia, aplicando el Teorema 2.1 (de Liouville), se deduce que toda aplicación 1-cuasiconforme de  $S^n$  en  $S^n$  es de la forma indicada en el teorema.

### 2.3. Teoría ergódica

En esta sección, introducimos algunos conceptos básicos sobre ergodicidad y presentaremos un esquema de la demostración del siguiente teorema, clave una vez más en la prueba del Teorema de Mostow, y siguiendo [23].

**Teorema 2.9.** *Sea  $M$  una variedad hiperbólica compacta. El flujo geodésico en  $M$  es ergódico.*

Comencemos con algunas definiciones para ubicarnos, siendo  $G$  un grupo segundo numerable localmente compacto y  $(X, \mu)$  un espacio de medida  $\sigma$ -finito.

**Definición 2.10.** Una acción medible de  $G$  en  $X$  se dice que *preserva la clase de medida* si para todo subconjunto medible  $A$  de  $X$ ,

$$\mu(gA) = 0 \iff \mu(A) = 0$$

con  $g \in G$ . Diremos que  $G$  *preserva la medida* si para todo subconjunto medible  $A$  de  $X$ ,  $\mu(A) = \mu(gA)$ .

**Definición 2.11.** Una acción de  $G$  en  $X$  que preserve la clase de medida será *ergódica* si para cualquier subconjunto  $G$ -invariante  $A$  de  $X$ , o bien  $\mu(A) = 0$  o  $\mu(X - A) = 0$ .

Consideremos una acción de  $G$  en  $X$  que preserve la medida, entonces se tiene que  $G$  actúa de forma unitaria sobre  $L^2(X, \mu)$  de la forma

$$\begin{aligned} G \times L^2(X, \mu) &\rightarrow L^2(X, \mu) \\ (g, f) &\mapsto f \circ g^{-1}. \end{aligned}$$

**Proposición 2.12.** *Sea  $\mu(X)$  finita. Dada una acción de  $G$  en  $X$  que preserve la medida, las siguientes afirmaciones son equivalentes:*

- (1) *La acción es ergódica.*
- (2) *Cualquier elemento  $G$ -invariante de  $L^2(X, \mu)$  es constante en casi todo punto.*

*Demostración.* (  $\implies$  ) Suponiendo que  $f \in L^2(X, \mu)$  es una función  $G$ -invariante no constante en casi todo punto, entonces existiría un conjunto  $A = \{x \in X \mid f(x) > a\}$  para cierto  $a \in \mathbb{R}$  tal que  $A$  es  $G$ -invariante,  $\mu(A) > 0$  y  $\mu(X - A) > 0$ . Luego la acción no sería ergódica.

(  $\impliedby$  ) Si suponemos que la acción no es ergódica, por definición  $\mu(A) > 0$  y  $\mu(X - A) > 0$  para algún  $A \subset X$ . Luego la función característica  $\chi_A \in L^2(X, \mu)$  sería una función no constante  $G$ -invariante.  $\square$

A partir de ahora,  $G$  pasará a ser  $\mathbb{R}$  como grupo con la suma.

**Teorema 2.13** (von Neumann). *Dada una acción de  $\mathbb{R}$  en  $X$  que preserve la medida, denotaremos por  $F \subseteq L^2(X, \mu)$  el espacio de funciones invariantes por  $\mathbb{R}$ .  $F$  es un subespacio cerrado ya que  $\mathbb{R}$  actúa sobre  $L^2(X, \mu)$  de forma unitaria. Sea  $P$  la proyección ortogonal de  $L^2(X, \mu)$  sobre  $F$ . Si la acción es continua, se tiene que la integral*

$$\int_0^N t \cdot f \, dt$$

*está bien definida. Entonces, bajo estas condiciones, tenemos el siguiente resultado (considerando el límite bajo la norma  $L^2$ ):*

$$Pf = \lim_{N \rightarrow \infty} \frac{1}{N} \int_0^N t \cdot f \, dt$$

*Esquema de demostración.* La prueba completa y detallada se puede seguir en [23]. Consiste en los siguientes tres pasos:

- Probar que  $F = \{g - t \cdot g \mid g \in L^2(X, \mu), t \in \mathbb{R}\}^\perp$ .
- Demostrar el resultado del teorema para los elementos de  $F$  y del conjunto  $\{g - t \cdot g \mid g \in L^2(X, \mu), t \in \mathbb{R}\}$ .
- Extenderlo para cualquier  $f \in L^2(X, \mu)$ , utilizando que el subespacio generado por  $F$  y  $\{g - t \cdot g \mid g \in L^2(X, \mu), t \in \mathbb{R}\}$  es denso en  $L^2(X, \mu)$ .

$\square$

Abordemos ahora la definición de flujo geodésico y algunos conceptos relacionados. Sea  $M$  una variedad Riemanniana compacta y  $T_1M$  su fibrado tangente unitario. Se tiene que  $\forall(x, v) \in T_1M \exists \gamma_{x,v} : \mathbb{R} \rightarrow M$  geodésica única tal que  $\gamma_{x,v}(0) = x$  y  $\dot{\gamma}_{x,v}(0) = v$ .

**Definición 2.14.** El *flujo geodésico* en  $T_1M$  es la acción de  $\mathbb{R}$  en  $T_1M$  dada por:

$$\begin{aligned} \mathbb{R} \times T_1M &\rightarrow T_1M \\ (t, (x, v)) &\mapsto (\gamma_{x,v}(t), \dot{\gamma}(t)). \end{aligned}$$

Se puede definir una métrica Riemanniana sobre  $T_1M$ , llamada métrica de Liouville. Para ello, consideremos dos curvas  $\alpha(t) = (x(t), v(t))$  y  $\beta(t) = (y(t), w(t))$  en  $T_1M$  tales que  $\alpha(0) = \beta(0)$ . Dado un  $t$  fijo, denotaremos por  $\tilde{v}(t)$  al transportado paralelo de  $v(t)$  en el punto  $x(0)$  a lo largo de la curva  $x$ . De forma análoga, escribiremos  $\tilde{w}(t)$  para el transportado paralelo de  $w(t)$  en el punto  $y(0)$  a lo largo de la curva  $y$ . Definimos el producto interior de  $\dot{\alpha}(0)$  y  $\dot{\beta}(0)$  como:

$$\langle \dot{\alpha}(0), \dot{\beta}(0) \rangle = \left\langle \frac{d}{dt} \tilde{v}(t) \Big|_{t=0}, \frac{d}{dt} \tilde{w}(t) \Big|_{t=0} \right\rangle + \langle \dot{x}(0), \dot{y}(0) \rangle$$

donde se han tomado, en la parte derecha de la igualdad, los productos interiores bajo el tensor métrico de Riemman en  $M$ . Es un ejercicio sencillo probar que esta expresión define una métrica en  $T_1M$ . Su forma de volumen asociada define una medida, llamada medida de Liouville. Esta es invariante ante el flujo geodésico, demostración que se puede consultar en [28].

*Resumen de demostración del Teorema 2.9.* Por el flujo geodésico en  $M$  una variedad hiperbólica compacta, nos referimos al flujo geodésico en  $X = T_1M$  como ya fue definido. Sea  $\mu$  la medida de Liouville en  $X$ , se tiene que podemos aplicar el Teorema 2.13 al flujo geodésico en  $X$ . Para probar la ergodicidad, basta con demostrar que los elementos de  $F$  son constantes en casi todo punto. Es más, puesto que el espacio de funciones continuas  $C(X)$  es denso en  $L^2(X, \mu)$ , llegará con probar que  $\forall f \in C(X)$ ,  $P(f)$  es constante en casi todo punto.

Para ello, se utiliza que fijada una función de  $f \in C(X)$ , existe una sucesión de reales positivos  $N_i$  que tiende a infinito, de forma que

$$\begin{aligned} Pf(x) &= \lim_{i \rightarrow \infty} \frac{1}{N_i} \int_0^{N_i} t \cdot f(x) dt, \\ Pf(x) &= \lim_{i \rightarrow \infty} \frac{1}{N_i} \int_{-N_i}^0 t \cdot f(x) dt, \end{aligned}$$

en casi todo punto  $x \in X$ .

La idea ahora consiste en en considerar un levantamiento de  $f$  a una función  $\tilde{f}$  definida sobre la cubierta  $T_1\mathbb{H}^n$  de  $X$ . Y definir:

$$\begin{aligned} \tilde{f}^+((x, v)) &= \lim_{i \rightarrow \infty} \frac{1}{N_i} \int_0^{N_i} \tilde{f}((\gamma_{x,v}(t), \dot{\gamma}_{x,v}(t))) dt, \\ \tilde{f}^-((x, v)) &= \lim_{i \rightarrow \infty} \frac{1}{N_i} \int_{-N_i}^0 \tilde{f}((\gamma_{x,v}(t), \dot{\gamma}_{x,v}(t))) dt, \end{aligned}$$

donde  $\gamma_{x,v}$  denota la geodésica en  $\mathbb{H}^n$  con punto inicial  $x$  y velocidad inicial  $v$ . Estarán bien definidas y serán iguales en casi todo punto.

La demostración continúa demostrando que  $\tilde{f}^+$  y  $\tilde{f}^-$  son constantes en casi todo punto, como consecuencia  $P(f)$  será constante en casi todo punto y quedará probado que el flujo geodésico en  $X$  es ergódico.  $\square$

## 2.4. Teorema de Mostow

En esta sección demostraremos el teorema de rigidez de Mostow, que motiva la realización de la parte experimental de este trabajo. Seguiremos la notación y prueba presentada en [23], con aportes en algunos puntos. Comencemos introduciendo el enunciado del teorema y algunas de sus consecuencias.

**Teorema 2.15. (Mostow).** *Sean  $M$  y  $N$  dos variedades hiperbólicas compactas de dimensión  $n \geq 3$ . Si  $f : M \rightarrow N$  es una equivalencia de homotopía, entonces  $f$  es homótopa a una isometría de  $M$  a  $N$ .*

Este resultado fue extendido para variedades completas de volumen finito en 1971 por Prasad, pero nos limitaremos al caso compacto. El teorema tiene como consecuencia la rigidez sobre la geometría de las variedades hiperbólicas compactas para  $n \geq 3$ , de forma que no se puede transformar la estructura geométrica de la variedad mediante deformaciones continuas que preserven el tipo de homotopía; y por lo tanto, la métrica hiperbólica de la variedad queda determinada por su tipo de homotopía. Esto no es cierto en superficies hiperbólicas ( $n = 2$ ), donde una misma estructura topológica puede ser compatible con varias estructuras geométricas.

Siguiendo el esquema de [24], la demostración se lleva a cabo de esta manera. En primer lugar, la idea será levantar  $f$  a una aplicación  $\tilde{f}$  entre las cubiertas universales de  $M$  y  $N$ , y probar que  $\tilde{f}$  admite una extensión continua  $\bar{f}$  al borde en el infinito, cuya restricción al borde es un homeomorfismo. A continuación, tendremos que probar que este homeomorfismo es conforme, para deducir, en base a resultados anteriores, una isometría  $h$  entre las cubiertas universales de  $M$  y  $N$  con la misma extensión al borde que  $\tilde{f}$ . Finalmente, veremos como  $h$  induce una isometría entre  $M$  y  $N$  homótopa a  $f$ , concluyendo la demostración.

Sean  $f$ ,  $M$  y  $N$  en las condiciones del Teorema de Mostow (Teorema 2.15). Denotando por  $g$  a la inversa en homotopía de  $f$ , podemos asumir sin pérdida de generalidad que  $f, g \in C^\infty$ , ya que por teoremas de aproximación  $f$  y  $g$  son homótopas a funciones  $C^\infty$ . Si levantamos  $f$  y  $g$  a sus respectivas aplicaciones  $\tilde{f}$  y  $\tilde{g}$  entre las cubiertas universales de  $M$  y  $N$ , obtenemos los siguientes diagramas conmutativos.

$$\begin{array}{ccc}
\mathbb{D}^n & \xrightarrow{\tilde{f}} & \mathbb{D}^n \\
\downarrow & & \downarrow \\
M & \xrightarrow{f} & N
\end{array}
\qquad
\begin{array}{ccc}
\mathbb{D}^n & \xleftarrow{\tilde{g}} & \mathbb{D}^n \\
\downarrow & & \downarrow \\
M & \xleftarrow{g} & N
\end{array}$$

Si recordamos, nuestro primer objetivo es demostrar que  $\tilde{f}$  se puede extender de forma continua al borde y ver que su restricción al mismo es un homeomorfismo, que se recoge en el siguiente teorema.

**Teorema 2.16.** *Con la notación anterior,  $\tilde{f}$  admite una extensión continua  $\bar{f}$  a  $\overline{\mathbb{D}^n}$ . Además,  $\bar{f}|_{\partial\mathbb{D}^n}$  es un homeomorfismo.*

Previo a la demostración, introduciremos y probaremos una serie de lemas que serán utilizados en la demostración del Teorema 2.16.

**Lema 2.17.** *Existen reales positivos  $c_1$  y  $c_2$  tales que,  $\forall x_1, x_2 \in \mathbb{D}^n$*

$$\frac{1}{c_1} \cdot d(x_1, x_2) - c_2 \leq d(\tilde{f}(x_1), \tilde{f}(x_2)) \leq c_1 \cdot d(x_1, x_2)$$

donde  $d(x_1, x_2)$  denota la distancia hiperbólica.

*Demostración.* Por ser  $f$  una función suave ( $C^\infty$ ) definida sobre un compacto, tenemos que su diferencial  $df$  está acotada. Por estar  $f$  y  $\tilde{f}$  relacionadas por el levantamiento, se tiene  $d\tilde{f}$  también acotada. El razonamiento es análogo para  $d\tilde{g}$ . Esto implica que  $\tilde{f}$  y  $\tilde{g}$  son Lipschitz, es decir, existe un real positivo  $c_1$  (basta tomar uno válido para ambas funciones) tal que

$$\begin{aligned}
d(\tilde{f}(x_1), \tilde{f}(x_2)) &\leq c_1 \cdot d(x_1, x_2), \\
d(\tilde{g}(x_1), \tilde{g}(x_2)) &\leq c_1 \cdot d(x_1, x_2).
\end{aligned}$$

Ya tenemos la segunda desigualdad. Ahora, por definición de inversa de homotopía,  $g \circ f \simeq \text{id}_M$ . Tanto  $g \circ f$  como  $\text{id}_M$  son  $C^\infty$ , luego el teorema de aproximación de Whitney nos garantiza la existencia de una homotopía suave  $F : M \times [0, 1] \rightarrow M$  con  $F(y, 0) = (g \circ f)(y)$  y  $F(y, 1) = y$ . Definiendo  $\tilde{g}$  de forma conveniente, podemos levantar  $F$  a la homotopía  $\tilde{F}$  entre  $\tilde{g} \circ \tilde{f}$  e  $\text{id}_{\mathbb{D}^n}$ . Siguiendo el razonamiento anterior, como  $M \times [0, 1]$  es compacto, se tiene que  $dF$  es acotado, luego  $d\tilde{F}$  también lo es. Así, para algún real positivo  $c$ ,

$$d((\tilde{g} \circ \tilde{f})(x), x) \leq c$$

y así, usando la desigualdad triangular, se llega a que

$$\begin{aligned}
d(x_1, x_2) &\leq d(x_1, (\tilde{g} \circ \tilde{f})(x_1)) + d((\tilde{g} \circ \tilde{f})(x_1), (\tilde{g} \circ \tilde{f})(x_2)) + d((\tilde{g} \circ \tilde{f})(x_1), x_2) \\
&\leq d((\tilde{g} \circ \tilde{f})(x_1), (\tilde{g} \circ \tilde{f})(x_2)) + 2c \\
&\leq c_1 \cdot d(\tilde{f}(x_1), \tilde{f}(x_2)) + 2c,
\end{aligned}$$

donde se concluye la demostración tras tomar  $c_2 = 2c/c_1$ .  $\square$

De aquí en adelante, denotaremos mediante  $\gamma$  a una geodésica en  $\mathbb{D}^n$  y  $\pi_\gamma$  será la proyección ortogonal de  $\mathbb{D}^n$  en  $\gamma$ .

**Lema 2.18.** *Sea  $\gamma$  una geodésica en  $\mathbb{D}^n$ ,  $s$  un real positivo. Definimos el entorno tubular abierto de  $\gamma$  como*

$$N_s(\gamma) = \{x \in \mathbb{D}^n \mid d(x, \gamma) < s\}$$

Denotando el complementario de  $N_s(\gamma)$  en el disco como  $N_s(\gamma)^c = \mathbb{D}^n \setminus N_s(\gamma)$ . Entonces existe una constante  $c(s)$  que depende de  $s$  de forma que se cumple:  $\lim_{s \rightarrow \infty} c(s) = +\infty$  y además,  $\forall p, q \in \mathbb{D}^n \mid d(p, \gamma) = d(q, \gamma)$ , se tiene la desigualdad

$$d_{N_s(\gamma)^c} \geq c(s) \cdot d(\pi_\gamma(p), \pi_\gamma(q)),$$

donde  $d_{N_s(\gamma)^c}$  denota la distancia hiperbólica en  $N_s(\gamma)^c$ .

*Demostración.* En esta demostración, trabajaremos por comodidad en el modelo del semiespacio  $\mathbb{H}^n$ . Sin pérdida de generalidad, podemos asumir que  $\gamma$  es una geodésica vertical cuyos puntos en el infinito son el 0 y  $+\infty$ , ya que cualquier geodésica se puede transformar en una vertical encontrando la transformación de Möbius adecuada. De este modo, el entorno tubular abierto descrito es un cono en el espacio euclidiano.

Sea  $l$  una curva diferenciable cualquiera en  $N_s(\gamma)^c$  con extremos  $p$  y  $q$ , como la de la Figura 2.1. Podemos asumir que  $l$  está parametrizada en  $[0, 1]$  de forma que  $l(0) = p$  y  $l(1) = q$ , de forma que se expresa con sus  $n$  funciones coordenadas como

$$\begin{aligned} l : [0, 1] &\rightarrow N_s(\gamma)^c \\ t &\mapsto (l_1(t), \dots, l_n(t)). \end{aligned}$$

Ahora, sea  $\lambda$  la pendiente del cono, borde del entorno tubular  $N_s(\gamma)$ , tenemos para la longitud de  $l$ :

$$\begin{aligned} \text{Longitud}(l) &= \int_0^1 \frac{|dl(t)|}{l_n(t)} \geq \int_0^1 \frac{|dl(t)|}{\frac{\lambda}{\sqrt{1+\lambda^2}}|l(t)|} \geq \frac{\sqrt{1+\lambda^2}}{\lambda} \int_0^1 \frac{|l(t)| \cdot |l'(t)|}{|l(t)|^2} dt \\ &\geq \frac{\sqrt{1+\lambda^2}}{\lambda} \left| \int_0^1 \frac{\langle l(t), l'(t) \rangle}{|l(t)|^2} dt \right| \geq \frac{\sqrt{1+\lambda^2}}{\lambda} |\ln |p| - \ln |q||. \end{aligned}$$

Donde se ha usado la desigualdad de Cauchy-Schwarz sobre el producto escalar en la penúltima desigualdad. Las igualdades se alcanzan si y solo si la curva  $l$  es el segmento euclídeo  $l_0$  de la Figura 2.1, luego  $\text{Longitud}(l) \geq \text{Longitud}(l_0)$ .

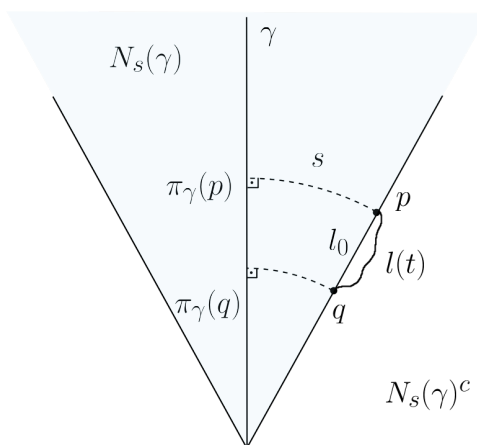


Figura 2.1: Idea gráfica para la demostración del Lema 2.18.

Finalmente, la proyección ortogonal trasladada el punto  $p$  a la geodésica vertical con una única coordenada no nula que toma el valor  $|p|$ , y lo mismo para  $q$ . Luego, es directo comprobar que la distancia entre las proyecciones ortogonales en  $\gamma$  será

$$d(\pi_\gamma(p), \pi_\gamma(q)) = |\ln |p| - \ln |q||$$

Así,  $c(s) = \frac{\sqrt{1+\lambda^2}}{\lambda}$  es la constante que buscábamos, donde la dependencia en  $s$  esta presente de forma implícita en la pendiente  $\lambda$ .  $\square$

**Lema 2.19.** Sea  $\varphi : \mathbb{D}^n \rightarrow \mathbb{D}^n$  una aplicación. Supongamos que existen reales positivos  $c_1$  y  $c_2$  tales que,  $\forall x_1, x_2 \in \mathbb{D}^n$

$$\frac{1}{c_1} \cdot d(x_1, x_2) - c_2 \leq d(\varphi(x_1), \varphi(x_2)) \leq c_1 \cdot d(x_1, x_2).$$

Entonces  $\forall \gamma$  geodésica en  $\mathbb{D}^n$ , existe una única geodésica  $\tilde{\gamma}$  en  $\mathbb{D}^n$  tal que  $d_H(\varphi(\gamma), \tilde{\gamma}) < \infty$ , distancia hiperbólica que está acotada por  $c_1$  y  $c_2$ . Además, si dos geodésicas  $\gamma$  y  $\gamma'$  tienen un mismo final,  $\tilde{\gamma}$  y  $\tilde{\gamma}'$  también (entendiendo por final a cada uno de los dos puntos del borde del infinito que determinan la geodésica).

*Demostración.* Una vez probada la existencia de  $\tilde{\gamma}$ , su unicidad es trivial ya que dos geodésicas distintas tienen al menos un punto en el infinito diferente, luego la distancia Hausdorff entre ellas es infinita. Trabajaremos en el modelo del semiespacio  $\mathbb{H}^n$ , manteniendo la notación para  $\varphi$  para denotar la aplicación correspondiente en este modelo.

Dados  $x, y \in \mathbb{H}^n$ , sin pérdida de generalidad,  $l_{xy}$  será la geodésica vertical que pasa por  $x$  e  $y$ , y  $\overline{xy}$  el segmento geodésico contenido en  $l_{xy}$  que une ambos puntos. Una vez más,  $N_s(l_{xy})$  será el entorno tubular abierto de distancia  $s$  de  $l_{xy}$ . El primer paso consiste en garantizar que la imagen por  $\varphi$  de  $\overline{xy}$  está dentro del entorno tubular abierto  $N_t(l_{\varphi(x)\varphi(y)})$  para un  $t$  correcto que solo depende de  $c_1$  y  $c_2$ .

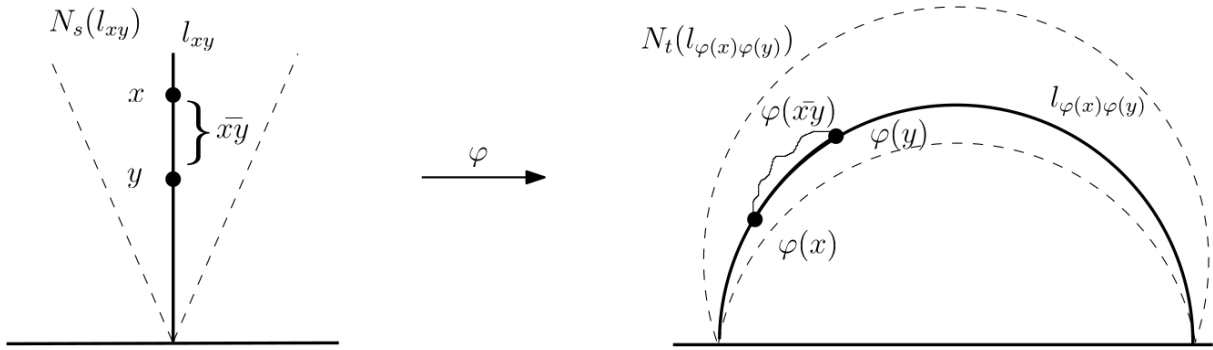


Figura 2.2: Idea gráfica sobre el primer paso de la demostración.

Tomemos en primer lugar un  $s$  cualquiera que cumpla, para  $c(s)$  descrito en el Lema 2.18,  $c(s) > c_1^2$ . Si  $\varphi(\bar{xy}) \subseteq N_s(l_{\varphi(x)\varphi(y)})$ , tomamos  $t = s$ . Sin embargo, se puede dar la situación de la Figura 2.3. En ese caso, tomamos cualesquiera  $a, b \in \bar{xy}$  tales que  $\varphi(a)$  y  $\varphi(b)$  pertenecen al borde de  $N_s(l_{\varphi(x)\varphi(y)})$  y que  $\varphi(\bar{ab})$  esté completamente contenido en  $N_s(l_{\varphi(x)\varphi(y)})^c$ . Aplicando el Lema 2.18, tendremos

$$\begin{aligned} \text{Longitud}(\varphi(\bar{ab})) &\geq c(s) \cdot d(\pi_{l_{\varphi(x)\varphi(y)}}(\varphi(a)), \pi_{l_{\varphi(x)\varphi(y)}}(\varphi(b))) \\ &\geq c(s) \cdot (d(\varphi(a), \varphi(b)) - 2s), \end{aligned}$$

donde la última desigualdad surge al aplicar la desigualdad triangular sobre la distancia entre los siguientes puntos en orden:  $\pi_{l_{\varphi(x)\varphi(y)}}(\varphi(a)) \rightarrow \varphi(a) \rightarrow \varphi(b) \rightarrow \pi_{l_{\varphi(x)\varphi(y)}}(\varphi(b))$  y tener en cuenta que la distancia entre un punto del borde del entorno tubular y su proyección a la geodésica será como mínimo  $s$ .

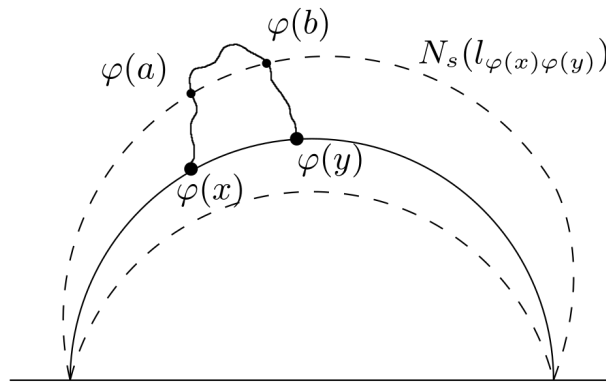


Figura 2.3: Caso en el que  $s$  no es suficientemente grande.

Por otra parte, aplicando la desigualdad del enunciado también obtenemos, considerando la caracterización de la longitud de la curva como el supremo en las particiones de la longitud de

la poligonal asociada a la partición, esto es:

$$\text{Longitud}(\varphi(\overline{ab})) = \sup_P \sum_{i=1}^n d(\varphi(t_{i-1}), \varphi(t_i)) \leq \sup_P \sum_{i=1}^n c_1 d(t_{i-1}, t_i) = c_1 \cdot d(a, b)$$

donde  $P$  denota la partición y  $\{t_i\}$  sus elementos, con  $t_0 = a$  y  $t_n = b$ . Combinando las desigualdades obtenidas hasta ahora, se llega a la siguiente cota,

$$d(\varphi(a), \varphi(b)) \leq \frac{2s \cdot c(s) + c_1^2 c_2}{c(s) - c_1^2}$$

donde fue necesario el requisito de tomar  $c(s) - c_1^2 > 0$  para garantizar el signo positivo en el denominador. Luego para cualquier punto  $p \in \overline{ab}$ ,

$$\begin{aligned} d(\varphi(p), l_{\varphi(x)\varphi(y)}) &\leq d(\varphi(p), \varphi(a)) + s \\ &\leq c_1 \cdot d(p, a) + s \\ &\leq c_1 \cdot d(b, a) + s \\ &\leq c_1(c_1 \cdot d(\varphi(a), \varphi(b)) + c_2) + s \\ &\leq c_1 \left( c_1 \cdot \frac{2s \cdot c(s) + c_1^2 c_2}{c(s) - c_1^2} + c_2 \right) + s, \end{aligned}$$

consiguiendo el  $t$  deseado cogiendo cualquier cota estrictamente mayor a la anterior, por ejemplo sumando 1, porque recordemos que el entorno tubular es abierto por definición.

Ahora, supongamos  $\gamma : \mathbb{R} \rightarrow \mathbb{D}^n$  es una parametrización normalizada de la geodésica. Por las hipótesis de este lema, se tiene que  $\varphi$  es propia; luego para  $x \rightarrow \infty$ , todos los puntos de acumulación (desde el punto de vista euclídeo) de  $\gamma(x)$  están en el borde del infinito  $\partial\mathbb{D}^n$ . Veamos que no puede haber dos puntos de acumulación distintos. Sean  $\{x_n\}$  y  $\{x'_n\}$  dos sucesiones que tienden a infinito tales que sus respectivas sucesiones  $\{\varphi(x_n)\}$  y  $\{\varphi(x'_n)\}$  tienden a dos puntos diferentes en  $\partial\mathbb{D}^n$ . Bajo la métrica hiperbólica, fijado  $y \in \gamma$ , la intersección

$$N_t(l_{\varphi(y)\varphi(x_n)}) \cap N_t(l_{\varphi(y)\varphi(x'_n)})$$

está acotada uniformemente respecto a  $n$ . Pero se ha probado previamente que para cualquier  $n$ , se tiene  $\varphi(\overline{yx_n} \cap \overline{yx'_n}) \subseteq N_t(l_{\varphi(y)\varphi(x_n)}) \cap N_t(l_{\varphi(y)\varphi(x'_n)})$  lo cual nos lleva a una contradicción con que  $\varphi$  sea propia.

De este modo,  $\varphi(x)$  converge a un único punto en el espacio euclidiano cuando  $x \rightarrow \infty$ . Con un razonamiento análogo, se obtiene que  $\varphi(x)$  también converge para  $x \rightarrow -\infty$ . Así, tendremos que

$$\varphi(\gamma) \subseteq \limsup_{x \rightarrow \infty} N_t(l_{\varphi(-x)\varphi(x)})$$

Finalmente, basta tomar la geodésica  $\tilde{\gamma}$  determinada por sus dos puntos en el borde del infinito  $\lim_{x \rightarrow \infty} \varphi(x)$  y  $\lim_{x \rightarrow -\infty} \varphi(x)$  (que tenemos garantizado que son puntos distintos por la desigualdad del enunciado), cuya distancia Hausdorff con  $\varphi(\gamma)$  estará acotada.

Para probar ahora la segunda afirmación del enunciado, supongamos otra geodésica  $\gamma'$  con un final común a  $\gamma$ . Escogiendo la parametrización de  $\gamma'$  tal que  $d(\gamma(x), \gamma'(x)) \rightarrow 0$  cuando  $x \rightarrow \infty$ , aplicando la hipótesis inicial

$$\frac{1}{c_1} \cdot d(\gamma(x), \gamma'(x)) - c_2 \leq d(\varphi(\gamma(x)), \varphi(\gamma'(x))) \leq c_1 \cdot d(\gamma(x), \gamma'(x)),$$

luego  $d(\varphi(\gamma(x)), \varphi(\gamma'(x))) \rightarrow 0$  cuando  $x \rightarrow \infty$ . Entonces  $\tilde{\gamma}$  y  $\tilde{\gamma}'$  también tendrán un final común.  $\square$

Ahora supongamos  $\varphi$  en las condiciones del Lema 2.19. Para definir la extensión  $\bar{\varphi}$  de  $\varphi$  con dominio  $\overline{\mathbb{D}^n}$  basta definirla para los puntos de  $\partial\mathbb{D}^n$ . Sea  $p$  uno de esos puntos, consideremos cualquier geodésica  $\gamma$  de  $\mathbb{D}^n$  con  $p$  un final de ella. Así, tomaremos  $\varphi(p)$  como el final correspondiente de  $\gamma$ . Nuestro objetivo ahora es probar que  $\bar{\varphi}$  es continua, para ello introducimos el siguiente lema.

**Lema 2.20.** *Supongamos  $\varphi$  como en el Lema 2.19. Sea  $H$  un hiperplano totalmente geodésico de dimensión  $(n - 1)$  de  $\mathbb{D}^n$  y sea  $l$  geodésica de  $\mathbb{D}^n$  ortogonal a  $H$ . Entonces existe  $c$  positivo que solo depende de  $c_1$  y  $c_2$  tal que*

$$\text{diam}(\pi_{\tilde{l}}(\varphi(H))) \leq c,$$

siendo  $\tilde{l}$  la única geodésica con  $d_H(\varphi(l), \tilde{l}) < \infty$ .

*Demostración.* (Seguir la figura para no confundirse con la notación.) Sean  $A$  y  $B$  los dos finales de  $l$ , y  $C$  un punto del borde del infinito de  $H$ . Denotemos por  $l_1$  a la geodésica con finales  $B$  y  $C$ , y por  $l_2$  a la que tiene finales  $A$  y  $C$ . Sean  $\tilde{l}_1$  y  $\tilde{l}_2$  las geodésicas a distancia finita de  $\varphi(l_1)$  y  $\varphi(l_2)$ , respectivamente. El punto  $x$  será la intersección de  $l$  y  $H$ , denotando por  $x_0$  a la proyección de  $\varphi(x)$  sobre  $\tilde{l}$ .

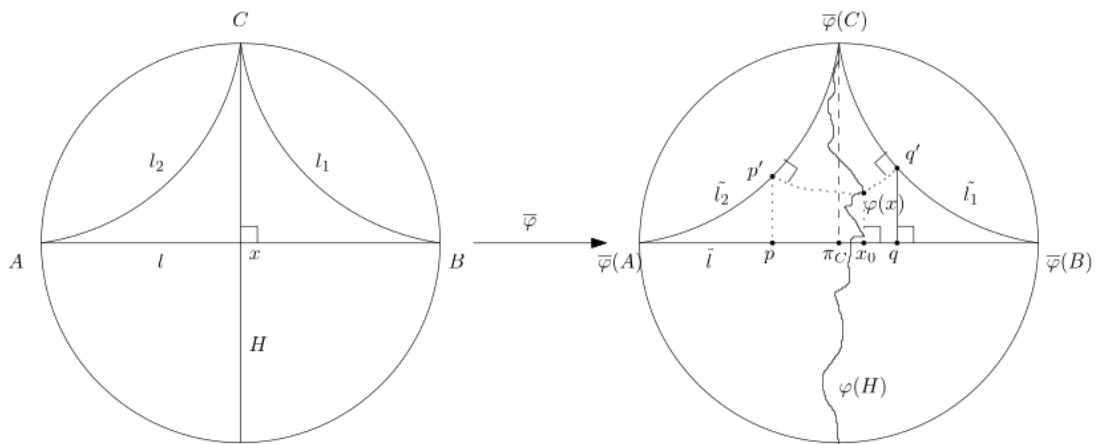


Figura 2.4

Sea  $d$  la distancia hiperbólica entre  $x$  y  $l_1$ , se tiene que  $d(\varphi(x), \varphi(l_1)) \leq c_1 d$ . El Lema 2.19 nos garantiza la existencia de  $t$  dependiente solo de  $c_1$  y  $c_2$  tal que

$$d_H(\tilde{l}_1, \varphi(l_1)) \leq t.$$

Usando la desigualdad triangular, llegamos a

$$d(\varphi(x), \tilde{l}_1) \leq c_1 d + t$$

y análogamente,  $d(\varphi(x), \tilde{l}_2) \leq c_1 d + t$ .

Sea  $p' = \pi_{l_2}(\varphi(x))$  y  $q' = \pi_{l_1}(\varphi(x))$ . Denotamos por  $p$  y  $q$  a las respectivas proyecciones de  $p'$  y  $q'$  sobre  $\tilde{l}$ . Sea  $\pi_C = \pi_{\tilde{l}}(\overline{\varphi}(C))$ , vemos que tanto  $x_0$  como  $\pi_C$  caen sobre el segmento que une  $p$  con  $q$ . Luego,

$$\begin{aligned} d(x_0, \pi_C) &\leq \max(d(x_0, p), d(x_0, q)) \\ &\leq \max(d(\varphi(x), p'), d(\varphi(x), q')) \\ &\leq c_1 d + t. \end{aligned}$$

Al haberlo probado para  $C$  arbitrario del borde del infinito de  $H$ , será válido para los puntos de  $H$ , es decir,  $\forall h \in \pi_{\tilde{l}}(\varphi(H))$  se tiene que  $d(x_0, h) \leq c_1 d + t$ . Quedando demostrado el lema.  $\square$

Ahora ya podemos probar el siguiente lema sobre la continuidad de  $\overline{\varphi}$ .

**Lema 2.21.** *Sea  $\varphi : \mathbb{D}^n \rightarrow \mathbb{D}^n$  como en el Lema 2.19 y consideremos la extensión  $\overline{\varphi} : \overline{\mathbb{D}^n} \rightarrow \overline{\mathbb{D}^n}$  definida anteriormente. Se tiene que  $\overline{\varphi}$  es continua.*

*Demostración.* Basta probar la continuidad en el borde. Sea  $P \in \partial\mathbb{D}^n$ , consideremos  $l$  una geodésica con final en  $P$  y  $\tilde{l}$  la geodésica cuya distancia a  $\varphi(l)$  es finita. Dado  $x \in l$ , denotemos por  $x'$  a la proyección de  $\varphi(x)$  sobre  $\tilde{l}$ . Consideramos el intervalo  $I_x$  contenido en  $\tilde{l}$  con centro en  $x$  y radio  $c$  (proporcionado por el Lema 2.20).

Ahora por el Lema 2.20,  $\varphi(H) \subseteq \pi_{\tilde{l}}^{-1}(I_x)$ . Y tal como hemos definido  $\overline{\varphi}$ , tenemos que cuando  $x \rightarrow \infty$  entonces  $x' \rightarrow \overline{\varphi}(P)$ . Ahora bajo la métrica euclidiana, a medida que  $x'$  tiende a  $\overline{\varphi}(P)$ , la banda  $\pi_{\tilde{l}}^{-1}(I_x)$  se va estrechando y también tiende uniformemente a  $\overline{\varphi}(P)$ . Luego  $\overline{\varphi}$  es continua en  $P$ , y por lo tanto continua en todo punto.  $\square$

Finalmente, juntando todos los lemas anteriores podremos demostrar el Teorema 2.16.

*Demostración del Teorema 2.16.* Por el Lema 2.17, tenemos que  $\tilde{f}$  cumple las hipótesis del Lema 2.19. Luego aplicando el Lema 2.21, sabemos que  $\tilde{f}$  se extiende de forma continua al borde del infinito. Tan solo nos queda probar que dicha extensión  $\overline{\tilde{f}}$  restringida al borde es un homeomorfismo.

Para ello, véase que  $\tilde{g}$  también puede ser extendida al borde de forma continua, usando un razonamiento análogo, extensión denotada por  $\overline{\tilde{g}}$ . En la demostración de 2.17, vimos que

$d((\tilde{g} \circ \tilde{f})(x), x)$  estaba acotada para una elección conveniente de  $\tilde{g}$ . Luego las extensiones continuas de  $\tilde{g} \circ \tilde{f}$  e  $id_{\mathbb{D}^n}$  coinciden cuando las restringimos al borde. Luego,  $\bar{f}|_{\partial\mathbb{D}^n}$  tiene inversa por la izquierda continua. Análogamente, si consideramos la misma idea para  $\tilde{f} \circ \tilde{g}$  llegamos a que tiene inversa por la derecha continua. Entonces, concluimos que  $\bar{f}|_{\partial\mathbb{D}^n}$  es un homeomorfismo.

Como demostración alternativa a la del artículo [23] de este último paso, podemos recordar que una aplicación continua, biyectiva definida de un espacio compacto a uno Hausdorff, automáticamente es un homeomorfismo. La continuidad ya fue probada, y la esfera del infinito es un espacio compacto y Hausdorff. Probemos que es biyectiva:

- Inyectividad: Dados dos puntos distintos  $x, y \in \partial\mathbb{D}^n$ , consideramos la geodésica  $\gamma$  con finales  $x$  e  $y$ , por definición,  $\bar{f}(x)$  y  $\bar{f}(y)$  son los correspondientes finales de la única geodésica con distancia finita a  $\varphi(\gamma)$ , y por lo tanto también serán puntos distintos.
- Sobreyectividad: Supongamos que no lo es para llegar a una contradicción. Entonces  $\bar{f}(\partial\mathbb{D}^n) \subsetneq \partial\mathbb{D}^n$ , espacio que por proyección estereográfica podemos llevar a  $\mathbb{R}^n$ . Por el teorema de Borsuk-Ulam [29], toda función continua  $h : S^n \rightarrow \mathbb{R}^n$  tiene un punto  $x \in S^n$  tal que  $h(x) = h(-x)$ . Como la proyección estereográfica es inyectiva, esto implicaría que  $\bar{f}|_{\partial\mathbb{D}^n}$  no es inyectiva. Llegando a una contradicción.  $\square$

Siguiendo con la idea presentada al comienzo para la demostración del Teorema de Mostow, el segundo paso consiste en probar que  $\bar{f}$  es conforme. Comenzamos con el siguiente lema.

**Lema 2.22.** *La aplicación  $\bar{f}|_{\partial\mathbb{D}^n}$  es cuasiconforme.*

*Demostración.* Usando la caracterización del Teorema 2.7, la aplicación será cuasiconforme si demostramos que  $\forall P \in \partial\mathbb{D}^n$ , se tiene que  $H(p, \bar{f})$  está acotada. Trabajemos en el modelo del semiespacio  $\mathbb{H}^n$  y sea  $P$  distinto a  $\infty$ . Por el teorema del punto fijo,  $\bar{f}$  tiene al menos un punto fijo; sin pérdida de generalidad, tomaremos  $\varphi(\infty) = \infty$ . Sea  $l$  la geodésica con finales  $P$  e  $\infty$ , y sea  $\tilde{l}$  la geodésica cuya distancia de Hausdorff con  $\bar{f}(l)$  es finita.

Si consideramos el hiperplano  $H$  perpendicular a  $l$ , aplicando el Lema 2.20, obtenemos que  $\text{diam}(\pi_{\tilde{l}}(\bar{f}(H))) \leq c$ . Luego tomando  $r$  como el ínfimo de la distancia entre  $\bar{f}(P)$  y la proyección sobre  $\tilde{l}$  de los puntos de  $\bar{f}(H)$ , obtenemos que  $\bar{f}(H)$  estará dentro de dos  $(n-1)$ -esferas de centro  $\bar{f}(P)$  y radios  $r$  y  $e^c r$  (donde el factor  $e^c$  traslada un punto verticalmente una distancia  $c$ ). Luego se tiene para  $P' \in \partial\mathbb{H}^n$ :

$$\frac{\sup_{|P'-P|=r} |\bar{f}(P') - \bar{f}(P)|}{\inf_{|P'-P|=r} |\bar{f}(P') - \bar{f}(P)|} \leq e^c \quad \forall r \geq 0.$$

Finalmente, como  $H(P, \bar{f})$  es invariante bajo transformaciones de coordenadas conformes, se tiene la acotación por  $e^c$  para todo  $P$  del borde bajo cualquier sistema de coordenadas conforme en un entorno de  $P$ , teniendo así que  $\bar{f}|_{\partial\mathbb{D}^n}$  es cuasiconforme.  $\square$

A continuación, usaremos que el flujo geodésico de  $M$  es ergódico, resultado del Teorema 2.9, para probar que  $\bar{f}|_{\partial\mathbb{D}^n}$  es 1-cuasi conforme. En [23] se prueba para dimensión  $n = 3$ , que será el caso que mostraremos. Para la prueba general con cualquiera dimensión, se puede consultar [27].

Consideremos  $n = 3$ , luego  $\mathbb{D}^3$  es una cubierta de  $M$ . Sea  $G$  el grupo de transformaciones de cubierta, la acción de  $G$  se extiende a  $\overline{\mathbb{D}^3}$ . Sea  $S_\infty^2$  el borde del infinito de  $\mathbb{D}^3$ , definimos la acción de  $G$  sobre  $S_\infty^2 \times S_\infty^2$  como

$$\begin{aligned} G \times (S_\infty^2 \times S_\infty^2) &\rightarrow S_\infty^2 \times S_\infty^2 \\ (g, (x_1, x_2)) &\mapsto (g \cdot x_1, g \cdot x_2) \end{aligned}$$

que es ergódica. Dada una geodésica  $l$ , para  $x, y \in l$  y dos vectores no nulos de los respectivos espacios tangentes,  $v_x \in T_x\mathbb{D}^3$  y  $v_y \in T_y\mathbb{D}^3$ , definimos el ángulo entre  $v_x$  y  $v_y$  como el ángulo entre  $v_y$  y el transportado paralelo de  $v_x$  en el punto  $y$ . Esta definición se extiende de forma continua para los dos finales de  $l$ . Sean  $p, q \in S_\infty^2$  tales finales,  $v_p \in T_p S_\infty^2$  y  $v_q \in T_q S_\infty^2$  no nulos, el ángulo entre  $v_p$  y  $v_q$  será el definido a lo largo de la geodésica.

Tenemos que  $\bar{f}|_{S_\infty^2}$  es diferenciable en casi todo punto, por ser  $\bar{f}$  cuasi conforme en  $S_\infty^2$ . Sea  $p \in S_\infty^2$ , tal que  $\bar{f}|_{S_\infty^2}$  es diferenciable en  $p$ , definimos el subespacio

$$V_p = \{v \in T_p S_\infty^2 \mid |d(\bar{f}|_{S_\infty^2})(v)| = |d(\bar{f}|_{S_\infty^2})| \cdot |v|\}.$$

Se tiene que  $\bar{f}$  es  $G$ -equivariante, luego  $\{V_p\}$  es invariante por  $G$ . La matriz de  $d\bar{f}|_{S_\infty^2}$  puede tener dos autovalores: (1) conjugados; (2) reales iguales diagonalizable; (3) iguales no diagonalizable; (4) reales distintos. Para (1) y (2), se tiene la 1-cuasi conformidad de  $\bar{f}|_{S_\infty^2}$  y  $\dim V_p = 2$ . Para (3) y (4), no sería conforme y  $\dim V_p = 1$ . Estas propiedades se preservan por equivarianza. Veamos que  $\dim V_p \neq 1$ : si  $\dim V_p = 1$ , tendríamos una distribución unidimensional en c.t.p. sobre  $S_\infty^2$ , así el ángulo entre cualesquiera direcciones de esta distribución sería constante en c.t.p. Si hacemos proyección estereográfica de  $S_\infty^2$  en  $\mathbb{R}^2$ , aplicación conforme, tendremos una nueva distribución de dim 1, denotada por  $\{V'_p\}_{p \in \mathbb{R}^2}$  en c.t.p. La condición de ángulo constante, se traduce en este caso a que el ángulo entre  $V'_p$  y  $r_{p,q}(V'_q)$  es constante c.t.p. para cualesquiera  $p, q \in \mathbb{R}^2$  en los que esté definido  $V'$ , donde  $r_{p,q}$  es la única reflexión de  $\mathbb{R}^2$  que intercambia  $p$  con  $q$ . Algo que es imposible, llegando a una contradicción y concluyendo que  $\bar{f}$  es 1-cuasi conforme en el borde.

Para concluir la demostración del teorema de Mostow, el Teorema 2.8 nos dice que  $\bar{f}$  es de hecho una aplicación conforme en el borde. En la sección sobre isometrías, se vio que entonces  $\exists h \in \text{Isom}(\mathbb{D}^n)$  con la misma extensión al borde que definimos para  $\bar{f}$ . Manteniendo la notación previa, ahora para el caso general de dimensión  $n$ , sea  $G$  el grupo de transformaciones de cubierta sobre la aplicación cubierta  $\mathbb{D}^n \rightarrow M$ . Tanto  $\bar{f}$  como  $h$  son  $G$ -equivariantes. Si consideramos la homotopía de  $\bar{f}$  a  $h$  mediante geodésicas normalizadas, entonces esta homotopía también será  $G$ -equivariante y por lo tanto reducible a  $M$ . Así, la isometría reducida de  $h$  es la que buscábamos, homótopa a  $f$ , quedando demostrado el Teorema 2.15.

## Capítulo 3

# Homología persistente en 3-variedades hiperbólicas

En este capítulo explicaremos todo lo relativo a la parte experimental de este trabajo. Conocidos los conceptos básicos sobre homología persistente y geometría hiperbólica, ahora combinaremos ambas áreas con el objetivo de calcular computacionalmente invariantes métricos, que son, de hecho, invariantes topológicos por el Teorema de Mostow, para lograr diferenciar variedades hiperbólicas de dimensión 3.

Nos centraremos exclusivamente en el caso compacto, que corresponde con el enunciado demostrado del Teorema 2.15, principalmente porque existen invariantes definitivos para distinguir variedades hiperbólicas no compactas de volumen finito. Una variedad hiperbólica compacta no posee puntos ideales en el borde del infinito. De aquí en adelante, siempre que hablemos de una variedad, nos estaremos refiriendo una 3-variedad hiperbólica compacta.

El invariante que utilizaremos serán los landscapes del Capítulo 1, para los que tenemos garantizada la estabilidad y el funcionamiento del bootstrap empírico para trabajar con muestras. Estos recogerán la información métrica de la variedad mediante el tiempo de nacimiento y muerte de las clases de homología persistente, comparando los landscapes promedio obtenidos para diferentes muestras de 3-variedades hiperbólicas, haremos un contraste de hipótesis respecto a la hipótesis nula de ser homeomorfas para un nivel de confianza establecido.

El lenguaje de programación utilizado será Python, que cuenta con multitud de librerías de uso libre que serán de gran ayuda en la creación del código. En particular, usaremos el módulo `SnapPy`, dedicado especialmente al estudio de 3-variedades hiperbólicas; la primera sección de este capítulo se dedicará a introducir las principales herramientas que usaremos de este programa, para familiarizar al lector con algunos conceptos nuevos. En la referencia [1], se encuentra el

enlace a su página web, que contiene la documentación detallada e información adicional sobre SnapPy.

El programa creado será capaz de realizar los siguientes pasos:

- Dada una 3-variedad hiperbólica compacta, obtener una muestra uniformemente distribuida bajo la métrica hiperbólica que represente la variedad original.
- Calcular el landscape asociado a la muestra.
- Repetir el proceso para  $N$  muestras y obtener, mediante bootstrap, una banda de confianza para el landscape de la variedad.
- Dadas dos variedades, realizar el procedimiento anterior y determinar mediante un contraste de hipótesis si no son homeomorfas/isométricas con cierto nivel de confianza.

En las próximas secciones, explicaremos cuidadosamente cada uno de los pasos, para entender con detalle como funciona el programa, invitando al lector la consulta del código presente en el Anexo II. Finalmente, mostraremos los resultados obtenidos para algunas variedades de interés.

### 3.1. Variedades en SnapPy

El módulo `SnapPy` cuenta con un gran número de variedades hiperbólicas de dimensión 3, de las cuales está registrada información de interés que describe la variedad o la relaciona con otras. La clase principal del módulo es `Manifold`, que da acceso a un amplio censo de variedades; en particular, usaremos el de variedades compactas orientables.

Dada una variedad  $M$ , como las descritas en el anterior párrafo, podremos obtener instantáneamente muchos invariantes, como su volumen, una presentación de su grupo fundamental, primer grupo de homología, etc. Además, tiene la opción de preguntar si dos variedades son isométricas; en el caso de variedades compactas, no siempre es capaz de dar una respuesta. Esta fue una de las motivaciones para tratar de distinguirlas mediante la homología persistente.

Otro comando da como salida el dominio de Dirichlet de la variedad, que no es más que una región fundamental en el espacio hiperbólico para la acción del grupo discreto de isometrías  $\Gamma$  cuyo cociente define la variedad. También se le llama dominio fundamental y constituye una teselación del espacio hiperbólico para dicho grupo. Formalmente, se define el dominio de Dirichlet  $D$  con centroide  $c$ , que tomaremos por defecto el origen, de la siguiente forma:

$$D = \{x \in \mathbb{H}^n \mid d(x, c) \leq d(x, \gamma c) \quad \forall \gamma \in \Gamma\},$$

donde  $d$  denota la distancia hiperbólica y  $\mathbb{H}^n$  el espacio hiperbólico. Véase que cada órbita de  $\Gamma$  corta al interior del dominio en a lo sumo un punto.

Una de las opciones disponibles permite visualizar el dominio fundamental contenido en la esfera de radio unidad. En la Figura 3.1a, se usa el modelo de Klein, luego los lados de los segmentos geodésicos que unen los vértices son líneas rectas, al contrario de lo que vemos en el modelo de Poincaré, representado en la Figura 3.1b para la misma variedad, donde los segmentos geodésicos son curvos.

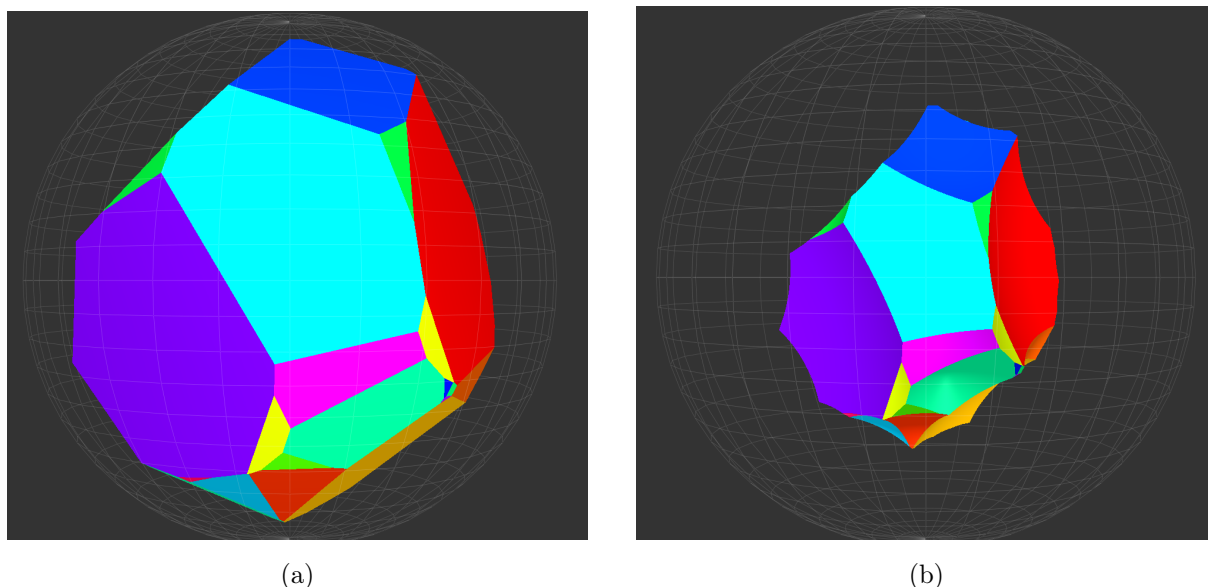


Figura 3.1: Dominio de Dirichlet bajo el modelo de: (a) Klein; (b) Poincaré.

En nuestro caso, el dominio fundamental estará descrito por un poliedro convexo con un número finito de caras. Tendremos acceso al conjunto de vértices, caras y aristas; las caras se identifican entre sí mediante generadores de la acción de  $\Gamma$ , pudiendo consultar la identificación concreta con simples comandos. Gráficamente, las caras identificadas se representan con el mismo color, haciendo más interesante la herramienta de visualización, la cual permite rotar el conjunto.

## 3.2. Muestreo aleatorio sobre una variedad

Dada una variedad, pretendemos generar una nube de puntos aleatoria sobre ella. Para ello, trabajaremos con su dominio de Dirichlet y los conjuntos de vértices, caras y aristas que lo determinan. El esquema que resume la idea detrás del programa encargado del muestreo es el siguiente:

- Obtener una triangulación del dominio fundamental.
- Calcular el volumen de los tetraedros y realizar un sorteo que seleccione cada tetraedro con

probabilidad proporcional al volumen.

- Subdivisión baricéntrica: para el tetraedro elegido, calcular su baricentro y generar una subdivisión en tetraedros con el baricentro como vértice común.
- Realizar otro sorteo para seleccionar un tetraedro bajo el mismo procedimiento.
- Iterar la subdivisión baricéntrica las veces necesarias para alcanzar el error deseado y seleccionar como punto final de la muestra el baricentro del último tetraedro.
- Repetir desde el segundo paso tantas veces como puntos queramos que tenga la muestra.

El truncamiento de este procedimiento define una sucesión de distribuciones que converge a la uniforme. Para obtener la triangulación, haciendo uso de la lista de caras obtenida con SnapPy, fijamos un vértice de referencia  $v_{ref}$  que estará contenido en todos los tetraedros construidos, en nuestro caso, tomamos el primer vértice de la primera cara. A continuación, recorremos todas las caras de la lista que no contengan dicho vértice de referencia; para cada una de las caras, volvemos a tomar el primer vértice como referencia  $v_0$  y triangulamos la cara como polígono bidimensional, considerando los vértices  $v_0, v_i, v_{i+1}$  para ir formando triángulos contiguos hasta llenar la cara, esto funciona computacionalmente ya que la lista de vértices que forman la cara se dan siguiendo un orden circular (horario o antihorario según la orientación establecida), garantizando que los triángulos no se intersequen. Finalmente, bastará tomar todas las triangulaciones de las caras construidas y considerar el tetraedro añadiendo  $v_{ref}$ . Es sencillo ver que esta construcción da lugar a una triangulación del dominio de Dirichlet. Este procedimiento se ilustra para un caso didáctico en la Figura 3.2.

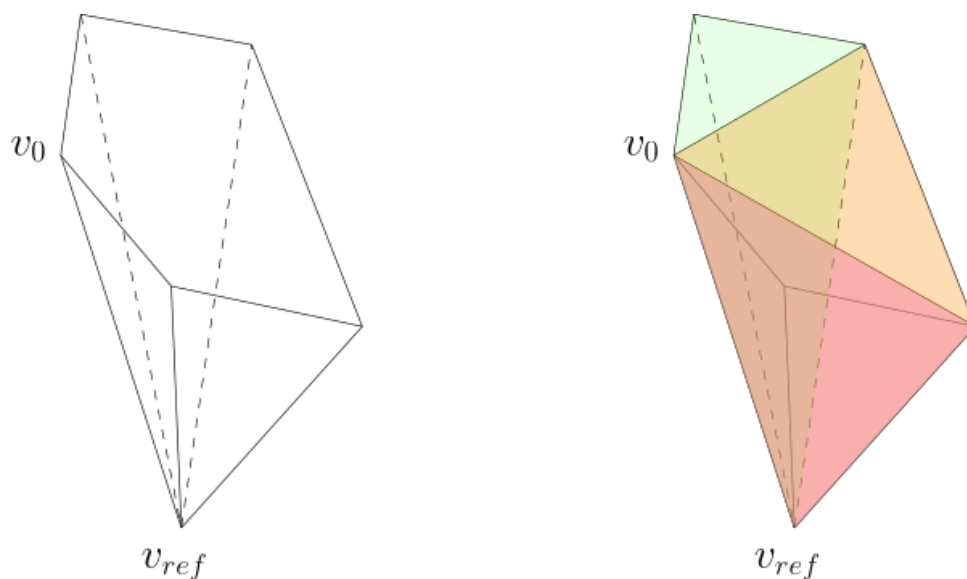


Figura 3.2: División en tetraedros para la triangulación del dominio de Dirichlet.

La triangulación obtenida será válida para cualquier modelo del espacio hiperbólico escogido. Por defecto, SnapPy devuelve las coordenadas de los vértices bajo el modelo de Klein. Sin ningún problema, se realiza la triangulación en este modelo, y una vez obtenidos todos los tetraedros, se pueden convertir a las coordenadas correspondientes al modelo que más nos convenga.

Conseguida la primera división en tetraedros, debemos obtener un método para calcular el volumen de cada tetraedro. Este problema se remonta a Poincaré y no existen fórmulas cerradas sencillas para el mismo. Generalmente, las fórmulas conocidas usan ángulos diedros pero para este trabajo hemos decidido usar la novedosa fórmula de la referencia [31] que solo usa las longitudes de las aristas, resumida a continuación.

Dados número complejos  $z, a_1, \dots, a_6$  se define la función compleja,

$$U = \text{Li}_2(z) + \text{Li}_2(a_1 a_2 a_4 a_5 z) + \text{Li}_2(a_1 a_3 a_4 a_6 z) + \text{Li}_2(a_2 a_3 a_5 a_6 z) \\ - \text{Li}_2(-a_1 a_2 a_3 z) - \text{Li}_2(-a_1 a_5 a_6 z) - \text{Li}_2(-a_2 a_4 a_6 z) - \text{Li}_2(-a_3 a_4 a_5 z),$$

donde  $\text{Li}_2$  es la función dilogaritmo, definida mediante la continuación analítica de la siguiente integral,

$$\text{Li}_2(x) = - \int_0^x \frac{\ln(1-t)}{t} dt$$

donde  $x$  es un número real menor que 1. La función dilogaritmo será implementada en el programa con la función `spence` del módulo `spicy.special`, que devuelve directamente el valor de  $\text{Li}_2$ , también llamado función de Spence, en el punto indicado.

Se define otra función de la forma

$$V = \frac{\sqrt{-1}}{4} \left\{ \left( U(a_1, a_2, a_3, a_4, a_5, z) \Big|_{z=z_-} - z_- \frac{\partial U}{\partial z} \Big|_{z=z_-} \cdot \ln(z_-) \right) \right. \\ \left. - \left( U(a_1, a_2, a_3, a_4, a_5, z) \Big|_{z=z_+} - z_+ \frac{\partial U}{\partial z} \Big|_{z=z_+} \cdot \ln(z_+) \right) \right\},$$

donde  $z_+$  y  $z_-$  son dos soluciones no triviales de la ecuación  $\text{Re} \left( z \frac{\partial U}{\partial z} \right) = 0$ , cuya expresión explícita se puede consultar en el artículo.

Consideremos un tetraedro hiperbólico  $T$  con sus vértices  $v_i$  y aristas etiquetadas como en la Figura 3.3, donde  $l_i$  denota la longitud de la arista correspondiente. Bajo esa notación, se tiene el siguiente teorema.

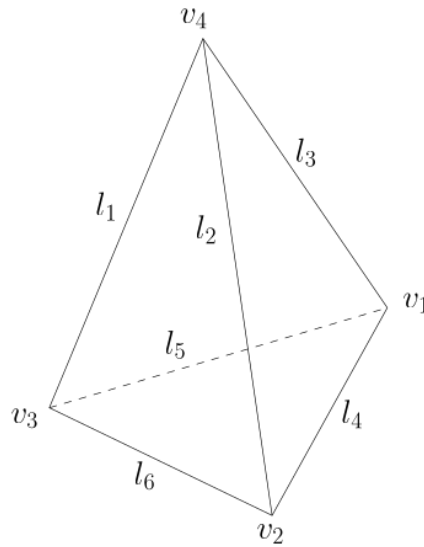


Figura 3.3: Disposición de los vértices y aristas para la fórmula de volumen.

**Teorema 3.1** (Teorema 2.2 en [31]). *El volumen del tetraedro hiperbólico  $T$  se puede expresar únicamente en función de las longitudes de las aristas que lo forman mediante la fórmula:*

$$\text{Volumen}(T) = V_l - \sum_{i=1}^6 l_i \frac{\partial V_l}{\partial l_i},$$

donde  $V_l = V(-e^{l_4}, -e^{l_5}, -e^{l_6}, -e^{l_1}, -e^{l_2}, -e^{l_3})$ .

Así, lo único que necesitaremos para calcular el volumen de cada uno de los tetraedros de la triangulación del dominio de Dirichlet será la longitud de las aristas geodésicas. En el programa, hacemos este cálculo usando el modelo del semiespacio superior, por ello tendremos que transformar las coordenadas de los tetraedros del modelo de Klein al de  $\mathbb{H}^n$ . La fórmula se validó correctamente en el cálculo del volumen hiperbólico a partir de triangulaciones de los dominios de Dirichlet y su comparación con el volumen del dominio arrojado por SnapPy.

Para continuar, debemos realizar el sorteo que otorgue a cada tetraedro una probabilidad proporcional al volumen que ocupa en el dominio. Para ello, definimos la probabilidad de escoger el tetraedro  $T_i$  como

$$P(T) = \frac{\text{Volumen}(T_i)}{\sum_{j=1}^{N_T} \text{Volumen}(T_j)}$$

donde  $N_T$  es el número total de tetraedros de la triangulación.

Ahora, asignaremos un segmento en el intervalo  $[0, 1]$  a cada tetraedro de longitud igual a su probabilidad, con intersección vacía. Generaremos un número aleatorio en  $[0, 1]$  con la función `random` y escogeremos el tetraedro correspondiente al subintervalo al que pertenezca el número aleatorio.

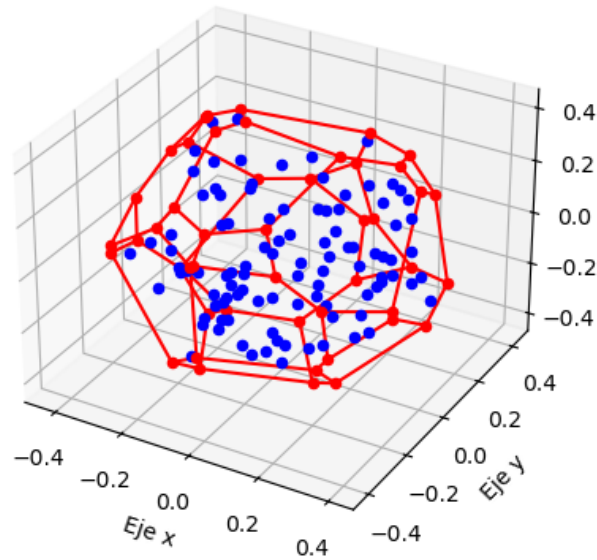


Figura 3.4: Muestra aleatoria de 100 puntos (azules) sobre el dominio de Dirichlet de una variedad. Los vértices y aristas del dominio se representan en color rojo bajo el modelo de Klein.

Escogido el tetraedro, el programa construido calcula su baricentro; usando este punto para trazar los tetraedros, se generan 24 tetraedros al tener en cuenta a mayores los vértices y baricentros de cada cara. Calculando el volumen de cada uno, podremos realizar de nuevo el proceso de elección para quedarnos con uno y repetir el procedimiento las veces que queramos, hasta seleccionar el último baricentro como primer punto de la muestra. Existen limitaciones en el programa debido a errores numéricos cuando el volumen de los tetraedros es del orden de  $10^{-16}$ , de forma que la fórmula de volumen puede llegar a fallar, incluso dando valores negativos. Para solucionarlo, colocamos un control que asigna volumen nulo a los tetraedros que se encuentran en esta situación, siempre y cuando aún existan tetraedros relativamente grandes en la división baricéntrica. Las implicaciones de esto son relevantes, para limitar sesgos producidos por este error numérico truncaremos el procedimiento en cuanto el diámetro de un tetraedro baje de  $10^{-4}$ . Así, las nubes de puntos generadas estarán a distancia de Hausdorff de  $10^{-4}$  de la muestra uniforme que se produciría sin truncamiento. Análogamente, se generarían el resto de puntos, dando lugar a una muestra aleatoria sobre el dominio de Dirichlet, como en el ejemplo de la Figura 3.4.

Al tomar un número de iteraciones finito y aproximaciones debido a errores numéricos para generar cada punto de la muestra, no se trata de una muestra totalmente uniforme. Es en este aspecto donde entra la estabilidad de los landscapes, garantizándonos que el landscape que obtendremos a partir de nuestra muestra obtenida con este método estará cerca del correspondiente a una muestra uniforme.

### 3.3. Cálculo de landscapes

Dada una muestra de puntos sobre el dominio de Dirichlet, queremos obtener su landscape asociado. Para ello, seguiremos los siguientes pasos:

- Obtener la matriz de distancias de la muestra  $M_d$ .
- Generar el diagrama de persistencia asociado a la filtración de Vietoris-Rips con el módulo `ripser` [2] a partir de  $M_d$ .
- Calcular el landscape correspondiente al diagrama de persistencia obtenido.

El primero de los pasos es uno de los más complejos técnicamente de introducir en el programa, especialmente por algunos problemas numéricos debido a la precisión con la que SnapPy da las coordenadas de los puntos. La matriz de distancias  $M_d$  recoge las distancias entre cualesquiera dos puntos de la variedad. Este cálculo no es trivial, ya que recordemos que las caras del dominio de Dirichlet están identificadas a pares, haciendo que la distancia entre dos puntos no tenga por qué corresponder con la distancia hiperbólica del camino más corto en el interior del dominio.

Haremos uso del Teorema del Poliedro de Poincaré, cuya prueba se puede ver en la referencia [32]. Nuestro dominio de Dirichlet está bajo las condiciones del teorema, esto nos garantiza que el grupo fundamental se genera con las isometrías que identifican las caras del dominio (información que es conocida con SnapPy). Debemos entonces encontrar dichas isometrías, que como ya se razonó en la sección 2.1, estarán representadas por elementos del grupo  $PSL(2, \mathbb{C})$ . Las transformaciones de Möbius están completamente determinadas por la imagen de tres puntos; es fácil ver que lo mismo ocurre con las isometrías.

**Proposición 3.2.** *Dados  $\{P_i\}_{i=\{1,2,3\}}$  y  $\{P'_i\}_{i=\{1,2,3\}}$  puntos de  $\mathbb{H}^3$  que determinen triángulos geodésicos congruentes, existe una única isometría  $f$  que preserva la orientación tal que  $f(P_i) = P'_i$  para  $i = \{1, 2, 3\}$ .*

*Demostración.* La existencia de al menos una isometría que preserve orientación y que lleva  $P_i$  en  $P'_i$  es un sencillo ejercicio de geometría hiperbólica, falta mostrar unicidad. Consideremos una superficie totalmente geodésica de  $\mathbb{H}^3$ , llamémosla  $H$ , que contenga los tres puntos  $P_i$ .  $H$  es isométrico con  $\mathbb{H}^2$  y port tanto, las isometrías de  $H$  son conjugadas a las isometrías de  $\mathbb{H}^2$ , que son transformaciones de Möbius y por tanto están determinadas por la imagen de tres puntos.

Supongamos que hay dos isometrías de  $\mathbb{H}^3$  que preserven orientación  $f$  y  $g$  que llevan  $P_i$  en  $P'_i$ . Se sigue que  $f^{-1} \circ g$  fija los puntos  $P_i$ , de hecho, fija la superficie totalmente geodésica  $H$  que los contiene. Por tanto  $f^{-1} \circ g$  define una isometría de  $H$  que es conjugada a una transformación de Möbius que fija tres puntos. Pero no existen isometrías distintas de la identidad que preserven orientación y fijen una superficie totalmente geodésica en  $\mathbb{H}^3$ , luego  $f = g$ .  $\square$

Recordemos que para cada matriz

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in PSL(2, \mathbb{C}),$$

se tiene la siguiente acción en el borde del infinito:

$$z \mapsto \frac{az + b}{cz + d} \quad z \in \mathbb{C},$$

que se extiende de forma única a una isometría en el espacio hiperbólico.

**Definición 3.3.** Un cuaternio  $q$  se expresa como  $q = a + bi + cj + dk$  con  $a, b, c, d \in \mathbb{R}$ . Cumplen la propiedad asociativa y distributiva, pero no son conmutativos. Se tiene  $i^2 = j^2 = k^2 = -1$  e  $ij = k$ , de donde se sacan el resto de relaciones.

La forma más sencilla de representar esta isometría para trabajar con ella computacionalmente es usando cuaternios en el modelo del semiespacio  $\mathbb{H}^3$ , cada punto es de la forma  $q^H = q_1 + q_2i + q_3j$ , con  $q_3 > 0$ . La isometría en  $\mathbb{H}^3$  actúa de la siguiente manera:

$$q^H \mapsto (aq^H + b)(cq^H + d)^{-1},$$

luego dados tres vértices  $\{P_i\}_{i=1,2,3}$  de una cara y sus tres imágenes  $\{P'_i\}_{i=1,2,3}$  en la cara identificada, obtenemos 3 ecuaciones en cuaternios que dan 12 ecuaciones (una por cada parte real, en  $i, j$  y  $k$ ) para 8 incógnitas (parte real e imaginaria de  $a, b, c, d \in \mathbb{C}$ ), con la condición adicional  $ad - bc = 1$ .

$$aP_i + b = P'_i(cP_i + d) \quad i = 0, 1, 2$$

El número de ecuaciones se puede reducir al darse cuenta que el complejo  $b$  no tiene parte en  $j$  ni en  $k$ . Luego, considerando solo la parte en  $j$  y en  $k$  de las ecuaciones anteriores, se obtienen 6 ecuaciones lineales y 6 incógnitas, que son las partes reales e imaginarias de  $a, c$  y  $d$ , que vamos a resolver.

Este último sistema a resolver se trata de un sistema lineal indeterminado, ya que estamos resolviéndolo para  $a, b, c, d$  complejos arbitrarios sin pedirles que la matriz tenga determinante 1, luego cualquier múltiplo real de una solución también lo será. Por otro lado, el rango debe ser 5 por la unicidad de solución dado en la Proposición 3.2; pero el hecho de que los vértices tengan error numérico hace que los triángulos geodésicos no tengan exactamente el mismo área, convirtiendo el sistema en determinado. Para solucionar este problema, se adapta el método de Gauss-Jordan permitiendo tolerancia en el cero (orden de  $10^{-10}$  hasta  $10^{-7}$ ), i.e., anulando valores que están muy cercanos al cero; de este modo, conseguimos que se mantenga compatible indeterminado. Resuelto el sistema, se despeja  $b$  de una de las ecuaciones y conseguimos la isometría buscada. El error en la resolución del sistema hace que aparezcan errores numéricos al identificar las caras, estos son del orden de  $10^{-10}$  en el peor de los casos.

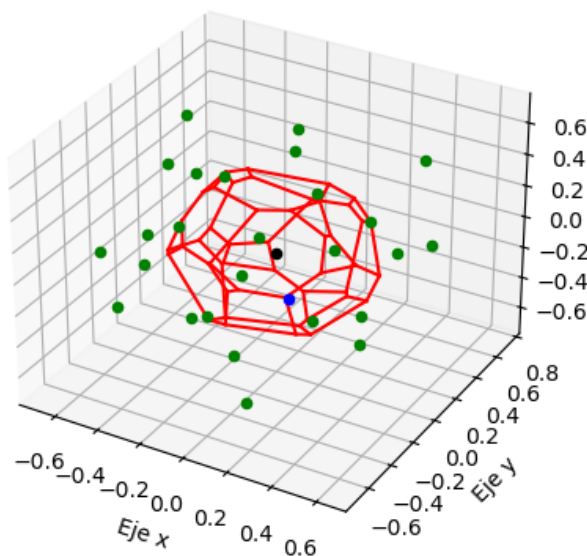


Figura 3.5: Cálculo de la distancia entre el punto negro y el azul, tomando todos los trasladados en verde de este último por las isometrías de las caras.

Continuando con el programa, ya conseguimos calcular todas las isometrías que identifican cada par de caras del dominio, las cuales generan el grupo fundamental. Sea  $I$  el conjunto de dichas isometrías, la distancia entre dos puntos  $p$  y  $q$  de la variedad, que asumimos que son del dominio fundamental, será

$$d(p, q) = \text{mín}\{d(p, i(q)) \mid i \in I\},$$

es decir, el mínimo entre las distancias de un punto y los trasladados del otro por las isometrías, como se observa en la Figura 3.5, siempre y cuando tengamos garantizada la Condición 3.4.

**Condición 3.4.** *Para que la fórmula de la distancia sea válida, debe ser necesario aplicar a lo sumo una vez cada isometría para rodear por completo el dominio fundamental, esta condición la comprobaremos geoméricamente como se ilustra a continuación.*

Véase el ejemplo de la Figura 3.6, en la izquierda observamos el dominio de Dirichlet en color rojo; en naranja, se representan las imágenes del dominio por cada una de las isometrías, las cuales rodean por completo el dominio fundamental. De este modo, como se ve a la derecha, dada una muestra de puntos dentro del dominio, esta estará completamente rodeada de una nube generada por las isometrías, permitiéndonos calcular la distancia entre dos puntos del dominio de Dirichlet sin tener que considerar más trasladados por composición de isometrías.

Trabajaremos exclusivamente con variedades que cumplan la condición anterior, realizando la comprobación manualmente para cada una de ellas. La idea es automatizar este proceso en

un futuro, creando un algoritmo que lo compruebe para la variedad introducida o, idealmente, obtener un teorema que de condiciones suficientes para que esto se cumpla.

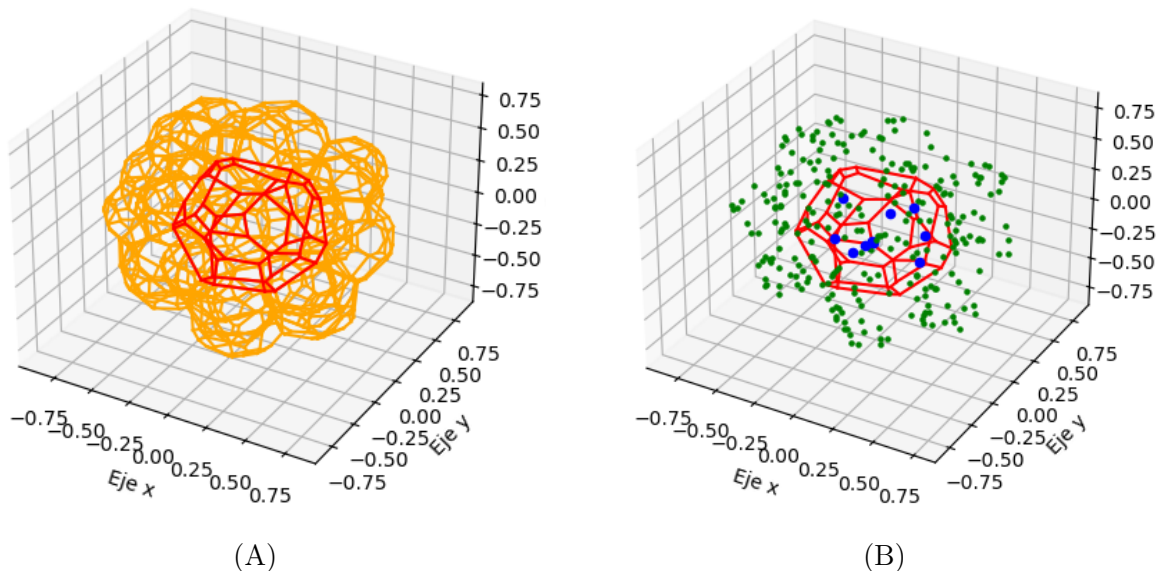


Figura 3.6: (A): Dominio de Dirichlet (rojo) y sus imágenes por las isometrías que identifican las caras (naranja). (B): Muestra de 10 puntos generada dentro del dominio de Dirichlet (azules) y nube de puntos imagen por las isometrías rodeando el dominio (verde).

Disponemos entonces de todos los elementos que componen la matriz de distancias  $M_d$ , que es esencial para la construcción de los complejos escogida, la de Vietoris-Rips. El módulo `ripser` funciona mediante la filtración de Rips, la cual consiste en la creación de los complejos de Vietoris-Rips como los de la Figura 1.2, usando el parámetro  $\epsilon$  del complejo de Vietoris-Rips  $VR(X, \epsilon)$  para definir la escala “temporal” que da cuenta de la persistencia de las clases de homología del complejo.

La función `ripser` recibe como entradas: la matriz de distancias de la muestra  $M_d$ , el grado máximo de homología que se desea calcular y el cuerpo  $\mathbb{Z}_p$  sobre el que se calcula la homología. Como salida, obtenemos el diagrama de persistencia correspondiente, como el de la Figura 3.7 (A), realizado a partir de una muestra de 100 puntos sobre la variedad de las anteriores figuras, hasta tercer grado de homología.

El teorema de rigidez es fundamental en este paso, ya que la filtración de Rips es dependiente de la métrica de la variedad. Como sabemos que dos variedades hiperbólicas homeomorfas serán isométricas, los landscapes promedio construidos bajo esta filtración deben ser iguales. Esto no ocurriría si no tuviésemos la rigidez de la métrica; por ejemplo, si consideramos en la métrica plana un toro  $\mathbb{T}_1$ , construido identificando los lados de un cuadrado de lado  $a$ , y otro  $\mathbb{T}_2$ , a partir

de un rectángulo de lados  $a/2$  y  $2a$ ; las clases persistentes nacerían en tiempos distintos y por lo tanto, los diagramas en  $H_1$  serían distintos, pese a ser homeomorfos, localmente isométricos e igual área.

En la práctica, solo generaremos diagramas de persistencia hasta segundo grado de homología. Para finalizar el cálculo de los landscapes de la muestra, simplemente tendremos que programar la fórmula que define el  $k$ -landscape a partir de los puntos del diagrama. Trabajaremos generalmente con los dos primeros landscapes asociados al diagrama de persistencia de cada grado de homología.

Véase la Figura 3.7 (B), donde se representan los 1-landscapes correspondientes a los diagramas de persistencia en  $H_1, H_2$  y  $H_3$ , manteniendo el código de colores. En el caso de  $H_0$ , hay una clase persistente que vive hasta el infinito (punto azul sobre el segmento horizontal discontinuo), la cual se elimina a efectos de calcular los landscapes asociados.

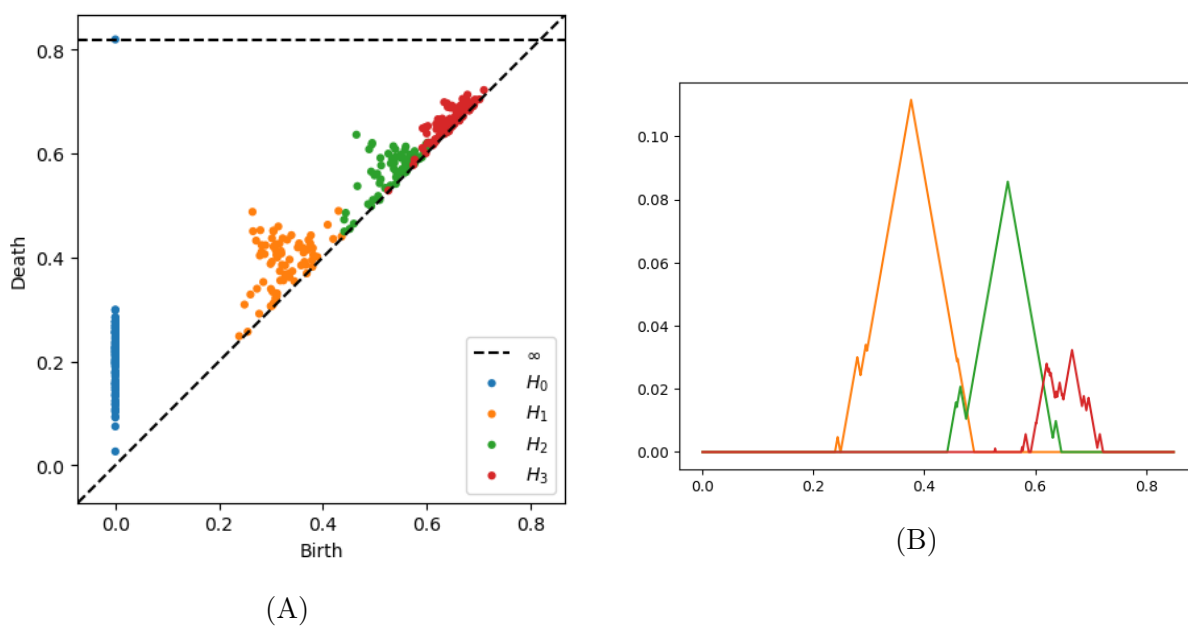


Figura 3.7: (A): Diagramas de persistencia. (B): 1-landscapes asociados.

### 3.4. Regiones de confianza mediante bootstrap

Replicaremos el procedimiento de las secciones anteriores para generar  $N$  muestras aleatorias de puntos sobre la variedad con sus correspondientes diagramas de persistencia y landscapes. De este modo, aplicaremos el bootstrap empírico del primer capítulo, obteniendo una banda de confianza en la que se encuentre el landscape promedio  $\mu$  de la variedad para un nivel de

significación dado. Recordemos que el intervalo de confianza será de la forma,

$$C_N = [\bar{\mathcal{L}}_N - q_\alpha, \bar{\mathcal{L}}_N + q_\alpha],$$

donde  $\bar{\mathcal{L}}_n$  es el promedio de los  $n$  landscapes muestreados y  $q_\alpha$  es el cuantil a confianza  $\alpha$ .

Para un número  $B$  de iteraciones de bootstrap, computaremos el cuantil correspondiente, obteniendo así una región de confianza como la de la Figura 3.8, la cual nos indica que contiene dentro el landscape de la variedad de donde generamos las muestras, con probabilidad  $\alpha$ . Usando el resultado de estabilidad de la Proposición 1.40, las franjas se deben ensanchar para dar cuenta del error entre la muestra generada y una muestra totalmente aleatoria uniforme.

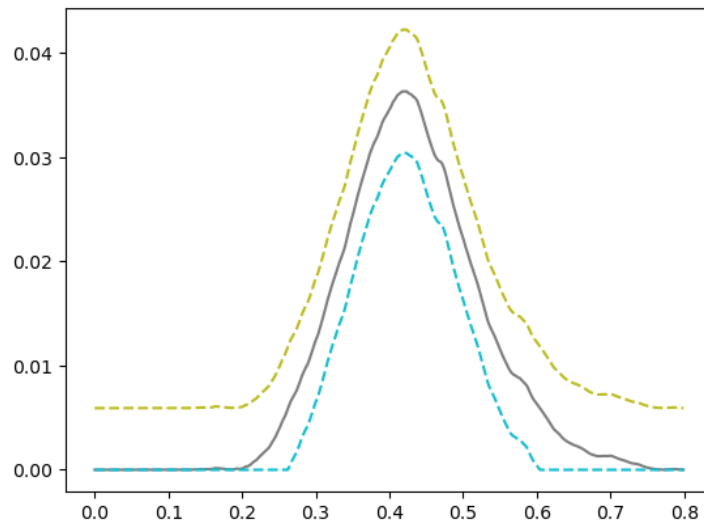


Figura 3.8: Banda de confianza para 1-landscape para  $H_1$  de la variedad  $m168(3,2)$  al 95% de confianza.

Con toda esta información, ya estamos en condiciones de comparar dos variedades. Fijado un landscape para un grado de homología concreto, si las variedades son homeomorfas, son isométricas por el Teorema de Mostow, luego sus landscapes deben ser iguales por tratarse de un invariante métrico. Generando las regiones de confianza para los landscapes a partir de las muestras, estableceremos un contraste de hipótesis donde la hipótesis nula es que sí son homeomorfas.

Supongamos que queremos ver si dos variedades  $M_1$  y  $M_2$  no son homeomorfas con confianza del 95%, la hipótesis a rechazar es que sí son homeomorfas. Para cada una de ellas para un número fijo de puntos de la muestra, ejecutamos el programa y obtenemos respectivamente los cuantiles  $q_1$  y  $q_2$ , para mismo  $N$ ,  $B$  y  $\alpha = 0,95$ . Sea  $\bar{\mathcal{L}}_1$  el landscape promedio empírico de la primera variedad y  $\bar{\mathcal{L}}_2$  el de la segunda, se tiene que

$$\|\bar{\mathcal{L}}_1 - \bar{\mathcal{L}}_2\|_\infty \leq \|\bar{\mathcal{L}}'_1 - \mu_1\|_\infty + \|\mu_1 - \mu_2\|_\infty + \|\mu_2 - \bar{\mathcal{L}}'_2\|_\infty + \left[ \|\bar{\mathcal{L}}_1 - \bar{\mathcal{L}}'_1\|_\infty + \|\bar{\mathcal{L}}_2 - \bar{\mathcal{L}}'_2\|_\infty \right],$$

donde  $\mu_i$  es el landscape promedio de cada variedad y  $\bar{\mathcal{L}}'_i$  el landscape promedio empírico obtenido si nuestra muestra fuese una muestra aleatoria uniforme sin truncamiento. Usando estabilidad (Proposición 1.40),  $\|\bar{\mathcal{L}}_i - \bar{\mathcal{L}}'_i\|_\infty \leq 2 \cdot 10^{-4}$  y  $\|\bar{\mathcal{L}}'_i - \mu_i\|_\infty \leq q_i + 2 \cdot 10^{-4}$  con confianza  $\alpha$ ,  $i = 1, 2$ . Por hipótesis  $\|\mu_1 - \mu_2\|_\infty = 0$ , luego diremos que no son homeomorfas a confianza  $\alpha$  cuando

$$\|\bar{\mathcal{L}}_1 - \bar{\mathcal{L}}_2\|_\infty > q_1 + q_2 + 8 \cdot 10^{-4}.$$

Intuitivamente, esto equivale a dibujar la región de confianza  $C = [\bar{\mathcal{L}}_2 - Q, \bar{\mathcal{L}}_2 + Q]$  con  $Q = q_1 + q_2 + 8 \cdot 10^{-4}$  y rechazar la hipótesis si la gráfica de  $\bar{\mathcal{L}}_1$  se sale de  $C$ .

### 3.5. Resultados

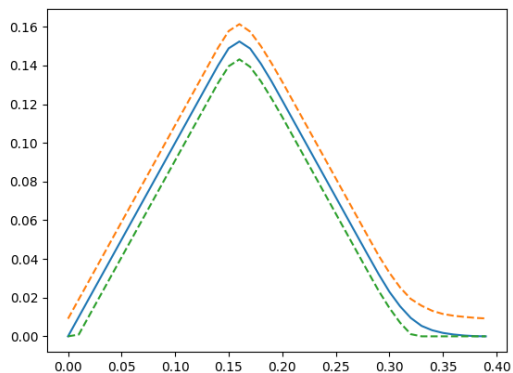
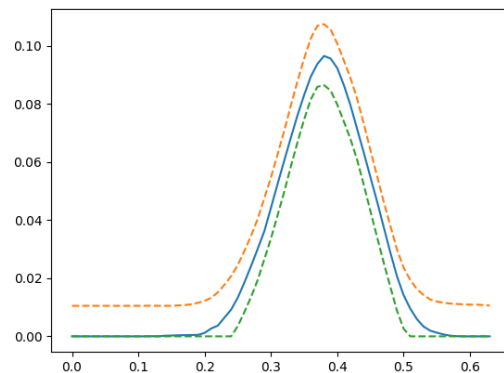
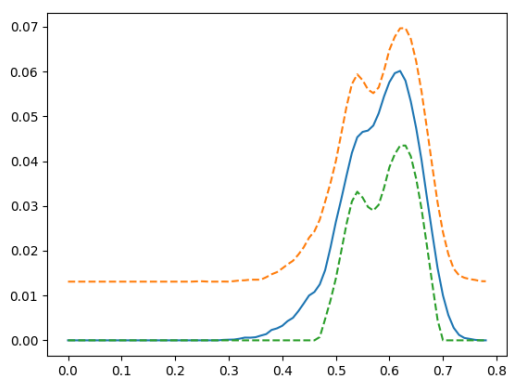
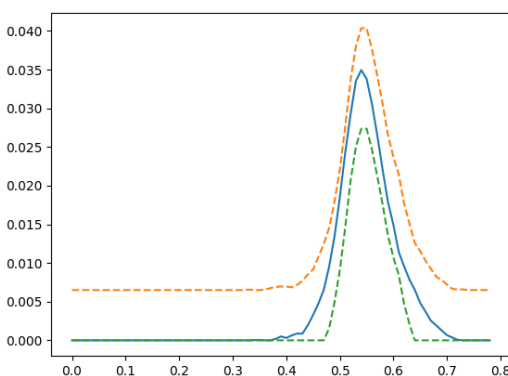
En esta sección presentamos algunos resultados de interés para distintas variedades, muchos otros ejemplos se pueden consultar en el Anexo I. Todos los que se presentan a continuación corresponden al análisis sobre 100 muestras de 100 puntos, aplicándoles un bootstrap de  $B = 100$  para  $\alpha = 95\%$  y usando homología con coeficientes en  $\mathbb{Z}_2$ , con la representación intuitiva anterior.

#### Grupos de control

Lo más importante del programa es que no tome decisiones equivocadas, es decir, si observamos que se rechaza la hipótesis nula para varios casos en los que las variedades son homeomorfas, estaríamos ante un error grave y este no serviría. Esta comprobación se realiza mediante grupos de control.

Para cada variedad  $M$ , realizaremos el contraste de hipótesis entre el dominio de Dirichlet de  $M$  centrado en el origen y otro dominio de  $M$  desplazado. Cambiar el centroide sobre el que se define el dominio fundamental, cambia su forma y, en general, no mantiene el número de caras del poliedro. Como son tomados de la misma variedad, no se debería rechazar la hipótesis de ser homeomorfas en ningún caso.

Por ejemplo, en la Figura 3.9, el landscape promedio empírico de la variedad m040(-4,3) para un dominio fundamental se encuentra dentro de las regiones de confianza construidas a partir del otro dominio. Sin embargo, en la Figura I.1 del anexo, para la variedad v3511(1,4), se obtiene que los grupos de control no son homeomorfos con confianza del 95%. Falla el primer landscape para  $H_2$ , pero viendo la imagen, se observa que el margen por el que se rechaza es muy pequeño. Lo mismo ocurre para otras variedades como la de la Figura I.2, las cuales tienen en común que son las de mayor volumen, luego el error numérico también se espera que sea mayor. Estos grupos nos sugieren que se deben realizar mejoras en el control de error del programa, por ejemplo aumentando el número de muestras y de bootstraps o mejorando la precisión en el muestreo. Los controles funcionaron correctamente para todas las variedades con volumen menor que 6 probadas.

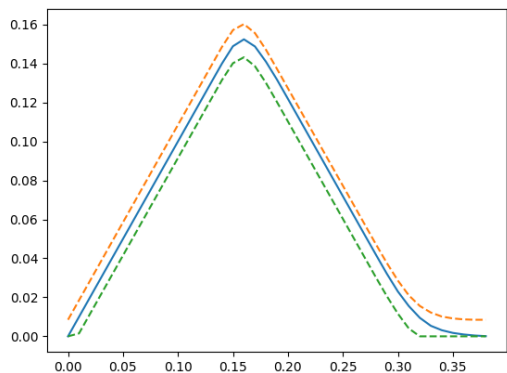
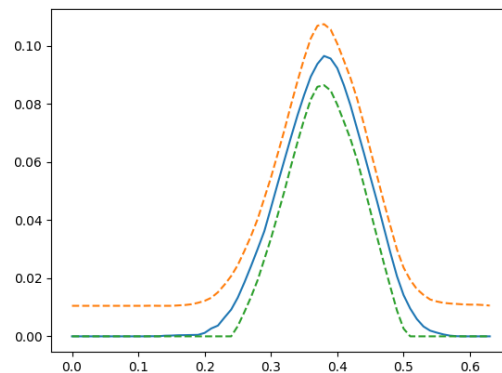
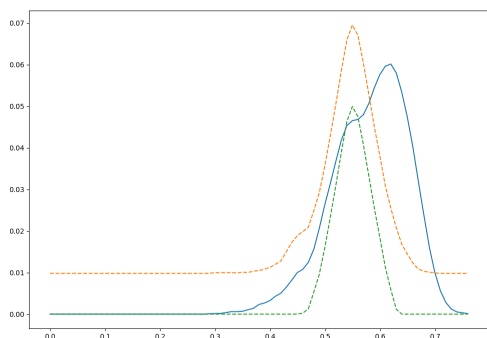
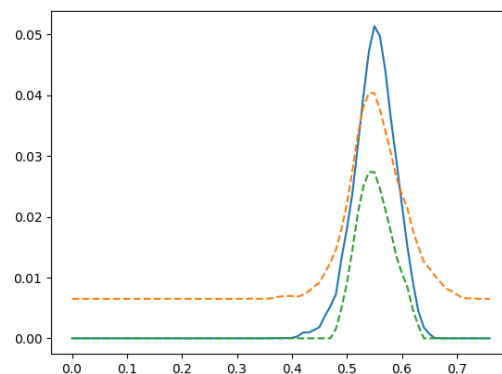
(a) Primer landscape (2-landscape) para  $H_0$ .(b) Primer landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .(d) Segundo landscape para  $H_2$ .Figura 3.9: Control con la variedad  $m040(-4,3)$ .

### Variedades no homeomorfas

La rigidez de la métrica hiperbólica garantiza que dos variedades homeomorfas siempre tienen el mismo volumen. Otro invariante topológico son los grupos de homología. Dadas dos variedades con volumen distinto u homología distinta, sabremos que no son homeomorfas; luego se espera que el programa sea capaz de afirmar que no lo son con confianza estadística  $\alpha$ . Para los ejemplos considerados en las Figuras I.3 (distinto volumen) y I.4 (mismo volumen con distinta homología), se obtuvo el resultado esperado, al igual que para muchos otros.

### Variedades con mismo volumen y mismo grupo de homología

El caso de mayor interés corresponde al de variedades con mismo volumen y homología, ya que en numerosas ocasiones, SnapPy no consigue dar una respuesta ante la pregunta de si son isométricas. Uno de estos casos es  $m040(-4,3)$  y  $m168(3,2)$ , cuyo análisis se ve en la Figura 3.10.

(a) Primer landscape (2-landscape) para  $H_0$ .(b) Primer landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .(d) Segundo landscape para  $H_2$ .Figura 3.10: Contraste para las variedades  $m040(-4,3)$  y  $m168(3,2)$ .

Nuestro programa distingue las variedades con  $H_2$ , resultando que se trata de variedades no homeomorfas con confianza estadística del 95%. Para estas variedades, los landscapes son significativamente diferentes y se descartan con bastante margen; sin embargo esto no siempre ocurre. Para las variedades  $m081(-4,1)$  y  $m116(-4,1)$  en la Figura I.5, el programa nos dice que no son homeomorfas al 95%, sin embargo el margen de descarte es muy pequeño. Esto implica que una modificación del programa que ensanche las regiones de confianza para dar cuenta de los errores numéricos en mejor medida, podría no distinguirlas. Esto mismo ocurre en la Figura I.6, con las variedades  $m294(1,3)$  y  $s090(5,1)$ .

En general, para mejorar en la toma de decisiones, es conveniente realizar más experimentos sobre otros cuerpos de coeficientes y con un bootstrap más profundo que mejore la precisión del cuantil.

## Capítulo 4

# Conclusiones

Este Trabajo de Fin de Grado representa una prueba de concepto que sienta las bases para un estudio más profundo y exhaustivo en el futuro sobre el cálculo de los invariantes métricos utilizados para distinguir variedades. Aunque el proceso de estimación teórica del landscape promedio es complejo, no es inviable. En principio, es posible obtener regiones con una confianza del 100%, lo que permitiría distinguir variedades de manera rigurosa y precisa.

Es evidente que se requieren más pruebas para mejorar y validar los resultados obtenidos. El tiempo dedicado a la depuración de errores ha limitado la presentación de un sumario más detallado de los resultados actuales. Para mejorar la precisión del método desarrollado, será necesario aumentar tanto el tamaño como el número de muestras, incrementar la cantidad de bootstraps y explorar más cuerpos de coeficientes, entre otros aspectos.

Además, otras mejoras potenciales incluyen el uso de diferentes observables persistentes, tales como el tiempo de vida medio, la entropía y el tiempo de vida máximo, para evaluar si ofrecen un mejor rendimiento. Recordemos que a efectos computacionales, nuestro landscape es un vector producto de la discretización de la función del landscape real, que es un invariante métrico; en particular, también lo serán otras magnitudes calculadas sobre ese vector. También es importante experimentar con otras filtraciones, lo cual justifica la variedad de filtraciones presentadas en el primer capítulo de este trabajo.

Una mejora crucial es establecer cotas específicas para todos los errores numéricos. Actualmente, solo se controla el error en el muestreo, pero no hay un control definido sobre la matriz de distancias, cuyo error crece con el volumen del dominio. Existe un error intrínseco asociado al uso de Snappy en la construcción del dominio de Dirichlet, que es del orden de  $10^{-8}$  en los vértices. En volúmenes grandes, estos vértices pueden estar próximos a la esfera del infinito, donde las isometrías experimentan mayores distorsiones y, por tanto, se incrementa el error numérico. Esto explica por qué el método es más efectivo para variedades con volúmenes pequeños y destaca

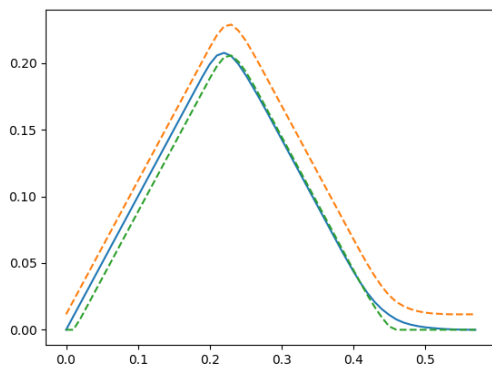
un área clave para futuras mejoras. Optimizado el programa de esta manera, se espera reducir significativamente el número de fallos en los contrastes de control para una misma variedad, dando mayor garantía de la veracidad de los resultados.

En resumen, queda abierta toda una línea de investigación, desde las estimaciones estadísticas y numéricas presentadas en este TFG hasta los resultados teóricos que se podrían obtener para distinguir variedades hiperbólicas bajo este invariante métrico.

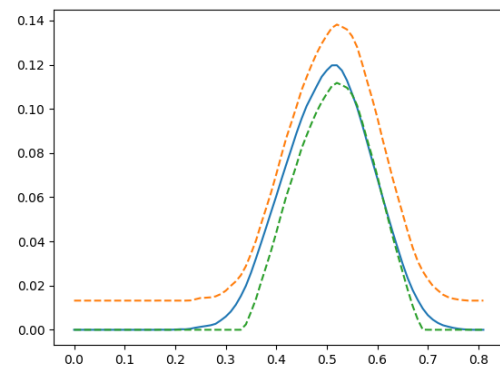
# Anexo I

## Contrastes para algunas variedades

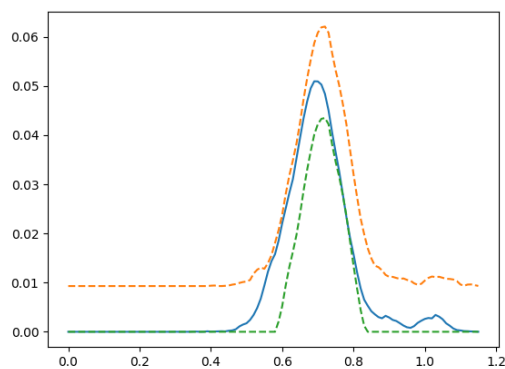
A continuación se muestran algunos de los resultados obtenidos mediante el programa:



(a) Primer landscape para  $H_0$ .



(b) Primer landscape para  $H_1$ .



(c) Primer landscape para  $H_2$ .

Figura I.1: Control para  $v_{3511}(1,4)$ .

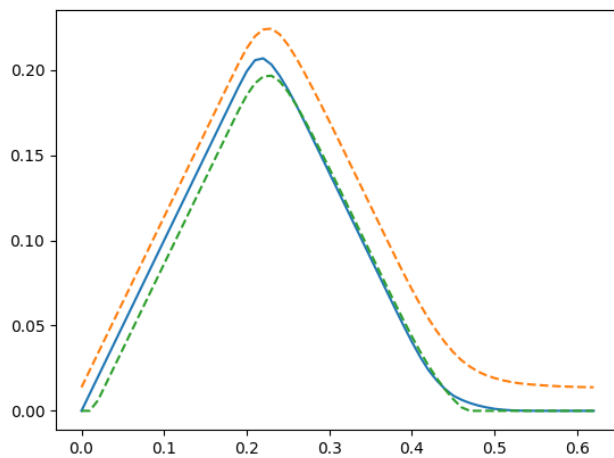
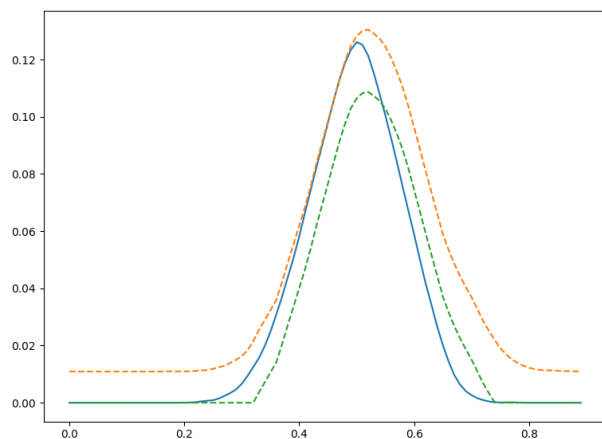
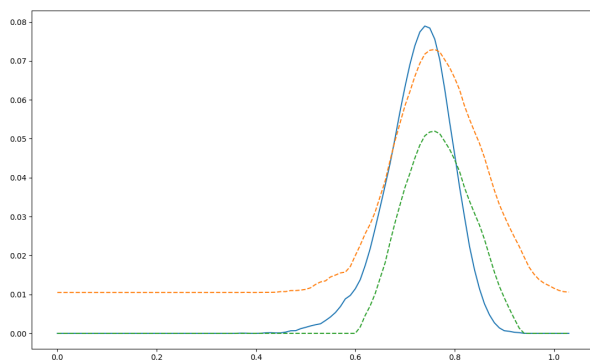
(a) Primer landscape para  $H_0$ .(b) Primer landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .

Figura I.2: Control para v3508(1,4).

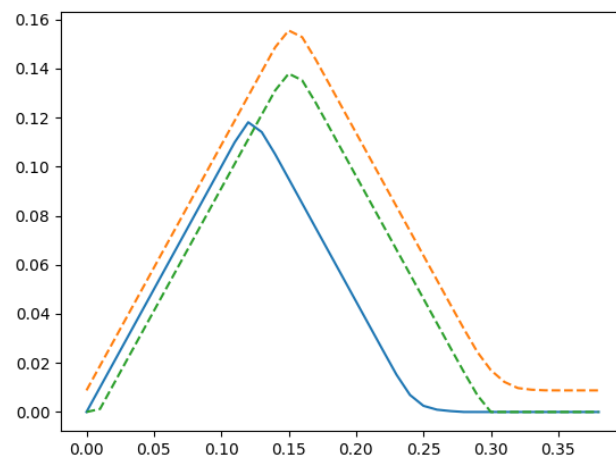
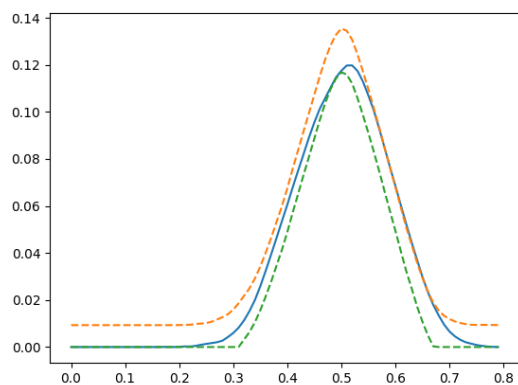
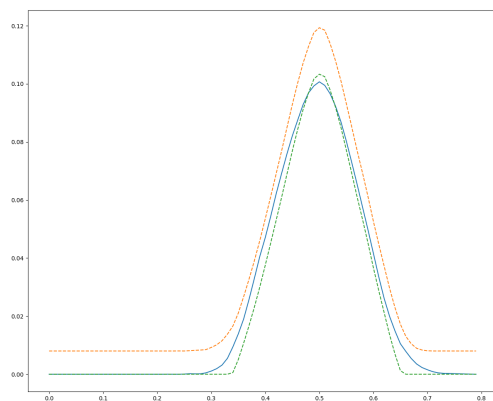
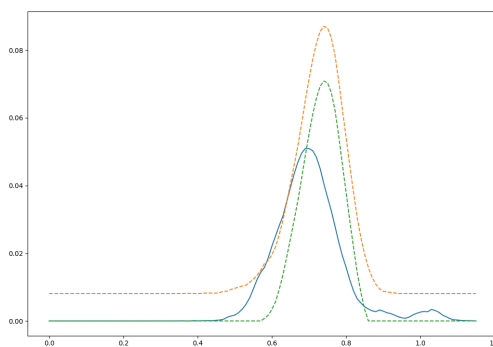
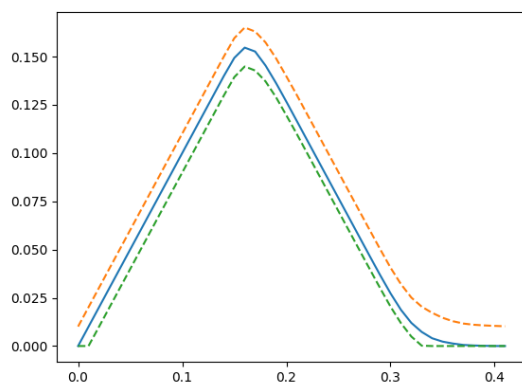
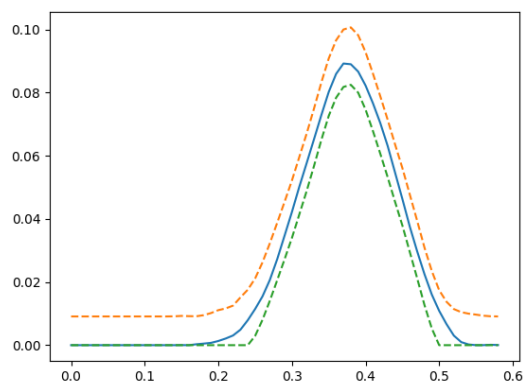
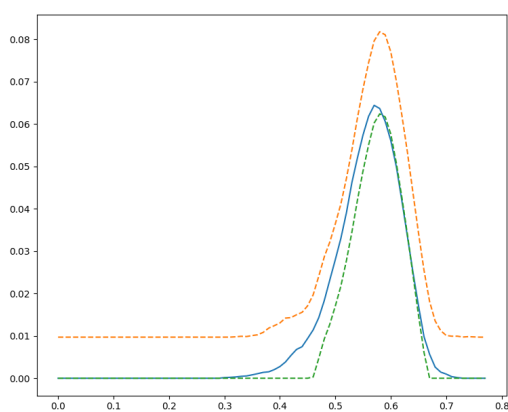
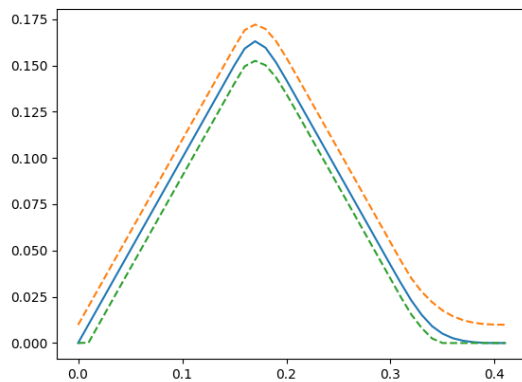
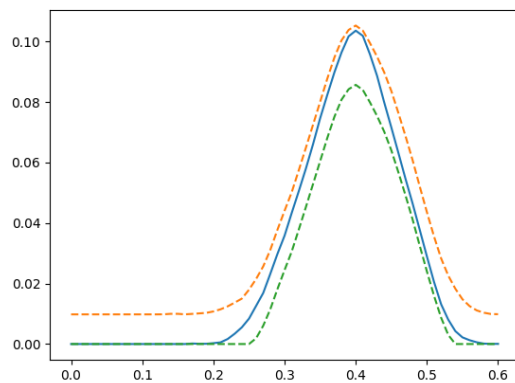
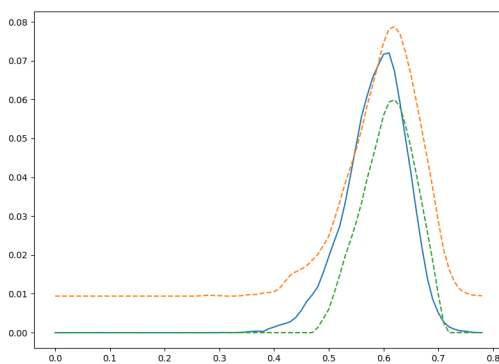


Figura I.3: Contraste para primer landscape asociado a  $H_0$  con las variedades  $m003(3,-4)$  y  $m040(-4,3)$  de volumen distinto.

(a) Primer landscape para  $H_1$ .(b) Segundo landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .Figura I.4: Contraste para  $v_{3508}(1,4)$  y  $v_{3511}(1,4)$ .

(a) Primer landscape para  $H_0$ .(b) Primer landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .Figura I.5: Contraste para  $m081(-4,1)$  y  $m116(-4,1)$ .

(a) Primer landscape para  $H_0$ .(b) Primer landscape para  $H_1$ .(c) Primer landscape para  $H_2$ .Figura I.6: Contraste para  $m_{294}(1,3)$  y  $s_{090}(5,1)$ .

## Anexo II

# Borrador del programa

El siguiente código ha sido realizado de forma conjunta y supervisada por el cotutor de este TFG:

**Carlos Meniño Cotón**, Departamento de Matemática Aplicada I, Universidade de Vigo.

No está explícito el programa que aplica el método de Gauss con tolerancia en el cero. Insistimos en que este código es por ahora solo un borrador, cuyo único fin es ilustrar la lógica computacional detrás del algoritmo, para el lector curioso.

### GENERADOR DE DIAGRAMAS

```
import numpy as np
#from scipy.spatial import Delaunay
import snappy as sp
from pyquaternion import Quaternion
#import volume_tetrahedra
import Hyperbolic_Distance_Disk_Model
import Random_Sampling_3D_Hyperbolic_Metric
import random as rd
import matplotlib.pyplot as plt
from ripser import ripser
from persim import plot_diagrams
import pandas as pd
import scipy
import Tolerance_Solve_Linear_Equations
from multiprocessing import Pool

##### VARIABLES #####
number_cores = 15

symbol = 'v3511(1,4)'
```

```

S = 100 #number of samplings
N = 100 #number of samples
eps = 0.0001 #sampling error
control = False

coeficient = 2
thresh = np.inf

M=sp.Manifold(symbol)
#D = M.dirichlet_domain()
D = M.dirichlet_domain(vertex_epsilon=10.0**-8, displacement = [0, 0, 0],
    centroid_at_origin=False, maximize_injectivity_radius=True)

#####

def Quaternionic_PSl2C_map(vertex_indices, vertex_image_indices): #
    p0 = Quaternion([np.real(vertex_indices[0][0]), np.imag(vertex_indices[0][0]),
        vertex_indices[0][1], 0])
    p1 = Quaternion([np.real(vertex_indices[1][0]), np.imag(vertex_indices[1][0]),
        vertex_indices[1][1], 0])
    p2 = Quaternion([np.real(vertex_indices[2][0]), np.imag(vertex_indices[2][0]),
        vertex_indices[2][1], 0])
    ip0 = Quaternion([np.real(vertex_image_indices[0][0]), np.imag(
        vertex_image_indices[0][0]), vertex_image_indices[0][1], 0])
    ip1 = Quaternion([np.real(vertex_image_indices[1][0]), np.imag(
        vertex_image_indices[1][0]), vertex_image_indices[1][1], 0])
    ip2 = Quaternion([np.real(vertex_image_indices[2][0]), np.imag(
        vertex_image_indices[2][0]), vertex_image_indices[2][1], 0])

    Coef_Matrix = [[-p0[2], 0, ip0[2], 0, ip0[0]*p0[2] + ip0[2]*p0[0], -(ip0[1]*p0[2]
        + ip0[2]*p0[1])],
        [-p1[2], 0, ip1[2], 0, ip1[0]*p1[2] + ip1[2]*p1[0], -(ip1[1]*p1[2]
        + ip1[2]*p1[1])],
        [-p2[2], 0, ip2[2], 0, ip2[0]*p2[2] + ip2[2]*p2[0], -(ip2[1]*p2[2]
        + ip2[2]*p2[1])],
        [0, -p0[2], 0, -ip0[2], ip0[1]*p0[2] - ip0[2]*p0[1], -ip0[2]*p0[0]
        + ip0[0]*p0[2]],
        [0, -p1[2], 0, -ip1[2], ip1[1]*p1[2] - ip1[2]*p1[1], -ip1[2]*p1[0]
        + ip1[0]*p1[2]],
        [0, -p2[2], 0, -ip2[2], ip2[1]*p2[2] - ip2[2]*p2[1], -ip2[2]*p2[0]
        + ip2[0]*p2[2]]]

    RM, RV, SOL, VAR = Tolerance_Solve_Linear_Equations.Gauss_Jordan(Coef_Matrix,
        [0,0,0,0,0,0], 1e-10) #1e-10 tolerance to 0

```

---

```

if VAR[0]!=[None]:
    a1 = np.array(VAR[0]).dot(np.ones(6))
else:
    a1 = 1

if VAR[1]!=[None]:
    a2 = np.array(VAR[1]).dot(np.ones(6))
else:
    a2 = 1

if VAR[2]!=[None]:
    d1 = np.array(VAR[2]).dot(np.ones(6))
else:
    d1 = 1

if VAR[3]!=[None]:
    d2 = np.array(VAR[3]).dot(np.ones(6))
else:
    d2 = 1

if VAR[4]!=[None]:
    c1 = np.array(VAR[4]).dot(np.ones(6))
else:
    c1 = 1

if VAR[5]!=[None]:
    c2 = np.array(VAR[5]).dot(np.ones(6))
else:
    c2 = 1

a = Quaternion([a1,a2,0,0])
d = Quaternion([d1,d2,0,0])
c = Quaternion([c1,c2,0,0])

b = -a*p0 + ip0*c*p0 + ip0*d
# print('Quaternionic error: ', b[2], b[3] )
b = Quaternion([b[0],b[1],0,0]) #rounding to 0 de j and k parts

# Tomar con determinante 1 es irrelevante a efectos de la accion:
det = a*d - b*c
#print(det[0])
if np.abs(det[1]) > 1e-8:
    print('Not real determinant', det) #No deberia entrar aqui nunca
if det[0] < 0:
    print('Determinante negativo') #No deberia entrar aqui nunca
    rdet = np.sqrt(-det[0])
    a = a*Quaternion([0,1,0,0])/rdet

```

```

    b = b*Quaternion([0,1,0,0])/rdet
    c = c*Quaternion([0,1,0,0])/rdet
    d = d*Quaternion([0,1,0,0])/rdet
else:
    rdet = np.sqrt(det[0])
    a = a/rdet
    b = b/rdet
    c = c/rdet
    d = d/rdet
#Control de errores
# Matriz = np.array([[a,b],[c,d]])
# Error_List = [Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_metric(
    PSL2C_Quaternionic_action_halfplane(Matriz, vertex_indices[i]),
    vertex_image_indices[i]) for i in range(len(vertex_indices))]
# if max(Error_List)>1e-10:
#     print('Error_List: ', Error_List)
return np.array([[a,b],[c,d]])

def best_face_map_v2(face_vertex_list, face_image_vertex_list):
    Matrix = []
    number_vertices = len(face_vertex_list)
    count = 0
    distort = np.inf
    for i in range(number_vertices):
        for j in range(i+1,number_vertices):
            for k in range(j+1,number_vertices):
                count = count + 1
                Map = Quaternionic_PSL2C_map([face_vertex_list[i],face_vertex_list
                    [j],face_vertex_list[k]],[face_image_vertex_list[i],
                    face_image_vertex_list[j],face_image_vertex_list[k]])
                M_i = [PSL2C_Quaternionic_action_halfplane(Map,p) for p in
                    face_vertex_list]
                dd = [Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_metric(
                    M_i[l], face_image_vertex_list[l]) for l in range(
                    number_vertices)]
                new_distort = max(dd)
                if new_distort < distort:
                    distort = new_distort
                    Matrix = Map
    return Matrix

def PSL2C_Quaternionic_action_halfplane(A,p): #A is PSL2C matrix whose complex
coordinates are given in quaternionic form, p is [z,h] with z complex and h>0
qp = Quaternion([np.real(p[0]),np.imag(p[0]),p[1],0])
action = (A[0][0]*qp + A[0][1])*(A[1][0]*qp + A[1][1]).inverse
if abs(action[3]) > 1e-10:
    print('Error: Parte k no nula', action[3])

```

---

```

    if action[2] < 0:
        print('Error: Parte j negativa')
    action = [action[0]+action[1]*1j , action[2]]
    return action

vertices=D.vertex_list()

tetraedros_klein=[]

verticeref=vertices[0]

faces = D.face_list()

#####Division en tetraedros #####

for i in range(D.num_faces()):
    verticescara=faces[i]['vertices']

    if verticeref not in verticescara:
        v0=verticescara[0]
        for j in range(1,len(verticescara)-1):
            tetraedro=[verticeref ,v0,verticescara[j],verticescara[j+1]]
            tetraedros_klein.append(tetraedro)
#pasar de modelo Klein a Poincare (s distancia al centro Klein, u distancia al
    centro Poincare)
#luego cada coordenada se debe corregir por una factor u/s=1/(1+raiz(1-s^2))

tetraedros_poincare=[]

for i in range(len(tetraedros_klein)):
    tetraedro=[]
    for j in range(4):
        s=np.linalg.norm(tetraedros_klein[i][j])
        tetraedro.append(np.array(tetraedros_klein[i][j])*1/(1+np.sqrt(1-s**2)))
    tetraedros_poincare.append(tetraedro)

#print(tetraedros_poincare)

#pasar de poincare a hiperplano

tetraedros_semiespacio=[]

for i in range(len(tetraedros_poincare)):
    tetraedro=[]
    for j in range(4):
        quaternion=Hyperbolic_Distance_Disk_Model.disk_to_halfspace(
            tetraedros_poincare[i][j])

```

```

        tetraedro.append([quaternion[0]+quaternion[1]*1j, quaternion[2]])
tetraedros_semiespacio.append(tetraedro)

vertices_poincare = []
for vertice in vertices:
    s = np.linalg.norm(np.array([vertice[0], vertice[1], vertice[2]]))
    vertices_poincare.append(np.array([vertice[0], vertice[1], vertice[2]])/(1+np.
        sqrt(1-s**2))) #vertices en el modelo de Poincare

vertices_halfspace = []
for vertice in vertices_poincare:
    p = Hyperbolic_Distance_Disk_Model.disk_to_halfspace(vertice)
    p = [p[0]+p[1]*1j, p[2]]
    vertices_halfspace.append(p)

##### Manifold Distance #####
print('Computing Relevant Group Elements')
distorsion_list = []
Faces_Vertices_Lists = []
Matrix_List = []
F = int(len(faces)/2)

Matrix_Aux_List = []
F = int(len(faces)/2)
for i in range(F):
    print(i)
    face = faces[2*i]
    v_1 = face['vertex_indices']
    v_i = face['vertex_image_indices']
    face_vertex_list = [vertices_halfspace[index] for index in v_1]
    face_image_vertex_list = [vertices_halfspace[index] for index in v_i]
    M = best_face_map_v2(face_vertex_list, face_image_vertex_list) #
        Quaternionic_PSL2C_map(face_vertex_list, face_image_vertex_list)
    Matrix_Aux_List.append(M)
    M_adjoint = [[M[1][1], -M[0][1]], [-M[1][0], M[0][0]]]
    Matrix_Aux_List.append(M_adjoint)
    Matrix = Matrix_Aux_List[2*i]
    M_i = [PSL2C_Quaternionic_action_halfplane(Matrix, p) for p in face_vertex_list
        ]
    dd = [Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_metric(M_i[k],
        face_image_vertex_list[k]) for k in range(len(v_1))]
    print(max(dd)) #, sum([ element**2 for element in dd])
    distorsion_list.append(max(dd)) #Maxima distorsion en los vertices de la cara

print('Relevant Generators Computed')

def hyperbolic_manifold_distance_semispace(p,q): #It is assumed that p,q belong to

```

---

```

    the fundamental domain, p,q given in semispace coordinates: [complex, positive
]
distance = Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_metric(p, q)
for Matrix in Matrix_Aux_List:
    r = PSL2C_Quaternionic_action_halfplane(Matrix, q)
    d = Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_metric(p, r)
    if d < distance:
        distance = d
return distance

def process_sampling(j):
    print('Sampling: ', j)
    List = []
    for i in range(N):
        #print('Sampling:', i)
        p = rd.random()
        cumulative = 0
        signal = False
        index = 0
        while signal == False:
            P = tetraedros_peso[index]
            cumulative = cumulative + P
            if p < cumulative:
                index_T = index
                signal = True
            else:
                index = index + 1
        Chosen_T = tetraedros_semiespacio[index_T]
        z = Random_Sampling_3D_Hyperbolic_Metric.hyperbolic_3D_sampling(Chosen_T
            [0], Chosen_T[1], Chosen_T[2], Chosen_T[3], eps)
        List.append(z)

#print('Computing Manifold Matrix Distance')

DM = np.zeros((N,N))
for i in range(N):
    for k in range(N):
        if i < k:
            DM[i][k] = hyperbolic_manifold_distance_semispace(List[i], List[k]
                ])
        elif k < i: #Ripser seems to work just with the upper triangular
            distande matrix, so this seems unnecessary
            DM[i][k] = DM[k][i]

#####Ripser#####
#print('Computing Persistence Diagram')
```

```

dgms = ripser(DM, maxdim=2, thresh = thresh, coeff=coeficient, distance_matrix
             =True) ['dgms'] #,thresh=thresh
#plot_diagrams(dgms, colormap='seaborn-pastel', show=True)

##### Saving diagrams H0 & H1 & H2
#####
dgms0 = dgms[0].tolist()
dgms0.remove(dgms0[-1]) #remove the last persitent class, that one dies at inf
      by rips construction
dgms1 = dgms[1]
dgms2 = dgms[2]
df0 = pd.DataFrame(dgms0, columns = ['Birth', 'Death'])
df1 = pd.DataFrame(dgms1, columns = ['Birth', 'Death'])
df2 = pd.DataFrame(dgms2, columns = ['Birth', 'Death'])
if control == False:
    if j<10:
        df0.to_csv('TU_CARPETA/Bootstrap-H0/'+symbol+'/DP-H0-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(0)+str(j)+'.csv')
        df1.to_csv('TU_CARPETA/Bootstrap-H1/'+symbol+'/DP-H1-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(0)+str(j)+'.csv')
        df2.to_csv('TU_CARPETA/Bootstrap-H2/'+symbol+'/DP-H2-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(0)+str(j)+'.csv')
    else:
        df0.to_csv('TU_CARPETA/Bootstrap-H0/'+symbol+'/DP-H0-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(j)+'.csv')
        df1.to_csv('TU_CARPETA/Bootstrap-H1/'+symbol+'/DP-H1-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(j)+'.csv')
        df2.to_csv('TU_CARPETA/Bootstrap-H2/'+symbol+'/DP-H2-'+symbol+'-Coef_Z
                  '+str(coeficient)+'-'+str(j)+'.csv')
    else:
        if j<10:
            df0.to_csv('TU_CARPETA/Bootstrap-H0/'+symbol+'_control/DP-H0-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(0)+str(j)+'.csv')
            df1.to_csv('TU_CARPETA/Bootstrap-H1/'+symbol+'_control/DP-H1-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(0)+str(j)+'.csv')
            df2.to_csv('TU_CARPETA/Bootstrap-H2/'+symbol+'_control/DP-H2-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(0)+str(j)+'.csv')
        else:
            df0.to_csv('TU_CARPETA/Bootstrap-H0/'+symbol+'_control/DP-H0-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(j)+'.csv')
            df1.to_csv('TU_CARPETA/Bootstrap-H1/'+symbol+'_control/DP-H1-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(j)+'.csv')
            df2.to_csv('TU_CARPETA/Bootstrap-H2/'+symbol+'_control/DP-H2-'+symbol
                      +'-Coef_Z'+str(coeficient)+'-'+str(j)+'.csv')
    return None

##### Random_Sampling #####

```

```

tetraedros_volumen = [Random_Sampling_3D_Hyperbolic_Metric.
    Volume_Tetrahedra_from_Vertices(T[0], T[1], T[2], T[3]) for T in
    tetraedros_semiespacio]
Volume = sum(tetraedros_volumen)
tetraedros_peso = [V/Volume for V in tetraedros_volumen]

pool = Pool(number_cores)
pool.map(process_sampling, range(S))

```

## MUESTREO ALEATORIO

```

import numpy as np
import math
import random as rd
import matplotlib.pyplot as plt
import volume_tetrahedra
# from scipy.optimize import fsolve

def conjugate(x): #conjugado de un punto en H3, x = (z,h), z complejo and h>0
    x[1] = -x[1]
    return x

def vectorize(x): # convertir (z,h) a 3D-array
    vector = np.array([np.real(x[0]), np.imag(x[0]), x[1]])
    return vector

def hyperbolic_3D_metric(x,y): #metrica hiperbolica en el semiespacio H3, los
    puntos son pares (z,h) con z complejo and h real
    vector_x = np.array([np.real(x[0]), np.imag(x[0]), x[1]])
    vector_y = np.array([np.real(y[0]), np.imag(y[0]), y[1]])
    vector_y_conj = np.array([vector_y[0], vector_y[1], -vector_y[2]])
    a = np.linalg.norm(vector_x-vector_y)/np.linalg.norm(vector_x-vector_y_conj)
    value = np.log((1+a)/(1-a))
    return value

def cos_angle(v,w):
    cos = abs(np.dot(v,w))/(np.linalg.norm(v)*np.linalg.norm(w))
    return cos

#####Interseccion de circunferencias
def func(x,a):
    #x, y = vars
    value =((x[0]-a[0][0])**2 + (x[1]-a[0][1])**2 + (x[2]-a[0][2])**2 - a[2]**2, (
        x[0]-a[1][0])**2 + (x[1]-a[1][1])**2 + (x[2]-a[1][2])**2 - a[3]**2, x[0]*a
        [4][0]+x[1]*a[4][1] - a[4][2])

```

```

return value

def intersection_ideal_spheres(a,ra,b,rb,plane_1,plane_2): #3ra coordenada de a y
b debe ser 0
A = np.array([[plane_1[0],plane_1[1]],[plane_2[0],plane_2[1]]])
B = np.array([plane_1[2],plane_2[2]])
sol_xy = np.linalg.solve(A,B)
vector_xy = np.array([sol_xy[0],sol_xy[1],0])
va = np.linalg.norm(vector_xy-a)
#vb = np.linalg.norm(vector_xy-b)
ha = np.sqrt(ra**2-va**2)
interseccion = [sol_xy[0]+ sol_xy[1]*1j, ha]
#hb = np.sqrt(rb**2-vb**2)
return interseccion

def intersection_ideal_sphere_with_sphere(a,ra,b,rb,plane_vector,approx): #3ra
coordenada de a debe ser 0
if abs(-b[0]+a[0]) > 1e-14:
    A = np.array([[2*(b[0]-a[0]),2*(b[1]-a[1]), 2*(b[2]-a[2])],[plane_vector
    [0],plane_vector[1],0],[b[2],0,-b[0]+a[0]]])
    B = np.array([ra**2-rb**2 + (b[0]**2-a[0]**2)+(b[1]**2-a[1]**2) + (b
    [2]**2-a[2]**2),plane_vector[2],a[0]*b[2]])
else:
    A = np.array([[2*(b[0]-a[0]),2*(b[1]-a[1]), 2*(b[2]-a[2])],[plane_vector
    [0],plane_vector[1],0],[0,b[2],-b[1]+a[1]]])
    B = np.array([ra**2-rb**2 + (b[0]**2-a[0]**2)+(b[1]**2-a[1]**2) + (b
    [2]**2-a[2]**2),plane_vector[2],a[1]*b[2]])
# print('sistema:',A,B)
sol_xyz = np.linalg.solve(A,B)
# print(sol_xyz)
va = np.linalg.norm(sol_xyz-a)
#vb = np.linalg.norm(sol_xyz-b)
ha = np.sqrt(ra**2-va**2)
#hb = np.sqrt(rb**2-vb**2)
A = np.array([[b[0]-a[0],b[1]-a[1],b[2]-a[2]],[plane_vector[0],plane_vector
[1],0],[plane_vector[1],-plane_vector[0],0]])
B = np.array([(b[0]-a[0])*sol_xyz[0] + (b[1]-a[1])*sol_xyz[1] + (b[2]-a[2])*
sol_xyz[2],plane_vector[2],0])
sol_uvw = np.linalg.solve(A,B)
normal = (sol_uvw - sol_xyz)/np.linalg.norm(sol_uvw - sol_xyz)
sol_1 = sol_xyz + ha*normal
sol_2 = sol_xyz -ha*normal
#sol_3 = sol_xyz +hb*normal
#sol_4 = sol_xyz -hb*normal
if np.linalg.norm(sol_1-approx) < np.linalg.norm(sol_2-approx):
    sol = sol_1
else:

```

---

```

        sol = sol_2
    return [sol[0]+sol[1]*1j, sol[2]]

def hyperbolic_sphere(x,r): #calcula centro y radio en la esfera euclidea
    correspondiente a la esfera hiperbolica centro x = (z,h) and radio r
    y = x[1]/np.sqrt(1-((np.exp(2*r)-1)/(np.exp(2*r)+1))**2)
    radius = y*(np.exp(2*r)-1)/(np.exp(2*r)+1)
    center = [x[0], np.sqrt((x[1])**2+radius**2)]
    return [center, radius]

def ideal_3D_center(x,y):
    if x[0] != y[0]:
        mod = np.abs(x[0]-y[0])**2
        t = 0.5*((y[1]**2-x[1]**2)/mod + 1)
        value = [x[0] + t*(y[0]-x[0]), 0]
    else:
        print('error: same complex part')
    return value

def mean_3D_point(x,y): # calcula el punto medio hiperbolico de un arco
    hiperbolico que junta los complejos x e y (sin vertices ideales)
    d = hyperbolic_3D_metric(x,y)
    if np.abs(x[0] - y[0]) < 1e-14:
        if x[1] < y[1]:
            center, radius = hyperbolic_sphere(x,d/2)
            mean_z = [center[0], center[1] + radius]
        else:
            center, radius = hyperbolic_sphere(y,d/2)
            mean_z = [center[0], center[1] + radius]
    else:
        z = [(x[0]+y[0])/2,(x[1]+y[1])/2]
        center, radius = hyperbolic_sphere(x,d/2)
        ideal_cent = ideal_3D_center(x,y)
        euc_radius = np.linalg.norm(vectorize(x)-vectorize(ideal_cent))
        plane_vector = [np.imag(x[0])-np.imag(y[0]),np.real(y[0])-np.real(x[0]),(
            np.imag(x[0])-np.imag(y[0]))*np.real(x[0]) + (np.real(y[0])-np.real(x
            [0]))*np.imag(x[0])]
        mean_z = intersection_ideal_sphere_with_sphere(vectorize(ideal_cent),
            euc_radius, vectorize(center), radius, plane_vector, vectorize(z)) #
            todavia algun problema en el calculo de punto medio (funcion mean_z)
    return mean_z

def barycenter(x,y,z): #baricentro de un triangulo hiperbolico en H3
    list_mean_points = [mean_3D_point(x,y),mean_3D_point(x,z),mean_3D_point(y,z)]
    if np.cross(vectorize(y)-vectorize(x),vectorize(z)-vectorize(x))[2] > 1e-14: #
        si no estan en un mismo plano vertical
        if np.abs(list_mean_points[0][0] - z[0])> 1e-14:

```

```

center_xyz = ideal_3D_center(list_mean_points[0], z)
radius_xyz = np.linalg.norm(vectorize(center_xyz)-vectorize(z))
v = list_mean_points[0]
plane_vector_1 = [np.imag(v[0])-np.imag(z[0]), np.real(z[0])-np.real(v
[0]), (np.imag(v[0])-np.imag(z[0]))*np.real(v[0]) + (np.real(z[0])-
np.real(v[0]))*np.imag(v[0])]
if np.abs(list_mean_points[1][0] - y[0]) > 1e-14:
    center_xzy = ideal_3D_center(list_mean_points[1], y)
    radius_xzy = np.linalg.norm(vectorize(center_xzy)-vectorize(y))
    v = list_mean_points[1]
    plane_vector_2 = [np.imag(v[0])-np.imag(y[0]), np.real(y[0])-np.
    real(v[0]), (np.imag(v[0])-np.imag(y[0]))*np.real(v[0]) + (np.
    real(y[0])-np.real(v[0]))*np.imag(v[0])]
    bar_1 = intersection_ideal_spheres(vectorize(center_xyz),
    radius_xyz, vectorize(center_xzy), radius_xzy, plane_vector_1,
    plane_vector_2)
else:
    bar_1 = [y[0], np.sqrt(radius_xyz**2 - np.abs(y[0]-center_xyz[0])
**2)]
else:
    center_xzy = ideal_3D_center(list_mean_points[1], y)
    radius_xzy = np.linalg.norm(vectorize(center_xzy)-vectorize(y))
    bar_1 = [z[0], np.sqrt(radius_xzy**2 - np.abs(z[0]-center_xzy[0])**2)]
else: #plane_vectors son paralelos en este caso, tomariamos como
plane_vector_2 el plano de interseccion de las esferas ideales
if np.abs(list_mean_points[0][0] - z[0]) > 1e-14:
    center_xyz = ideal_3D_center(list_mean_points[0], z)
    radius_xyz = np.linalg.norm(vectorize(center_xyz)-vectorize(z))
    v = list_mean_points[0]
    plane_vector_1 = [np.imag(v[0])-np.imag(z[0]), np.real(z[0])-np.real(v
[0]), (np.imag(v[0])-np.imag(z[0]))*np.real(v[0]) + (np.real(z[0])-
np.real(v[0]))*np.imag(v[0])]
if np.abs(list_mean_points[1][0] - y[0]) > 1e-14:
    center_xzy = ideal_3D_center(list_mean_points[1], y)
    radius_xzy = np.linalg.norm(vectorize(center_xzy)-vectorize(y))
    a = vectorize(center_xyz)
    b = vectorize(center_xzy)
    ra = radius_xyz
    rb = radius_xzy
    plane_vector_2 = [2*(b[0]-a[0]), 2*(b[1]-a[1]), ra**2-rb**2 + (b
[0]**2-a[0]**2)+ (b[1]**2-a[1]**2) + (b[2]**2-a[2]**2)]
    bar_1 = intersection_ideal_spheres(a, ra, b, rb, plane_vector_1,
    plane_vector_2)
else:
    bar_1 = [y[0], np.sqrt(radius_xyz**2 - np.abs(y[0]-center_xyz[0])
**2)]
else:

```

---

```

        center_xzy = ideal_3D_center(list_mean_points[1],y)
        radius_xzy = np.linalg.norm(vectorize(center_xzy)-vectorize(y))
        bar_1 = [z[0], np.sqrt(radius_xzy**2 - np.abs(z[0]-center_xzy[0])**2)]
# Control: Barycenter computed from the other intersections
# if list_mean_points[2][0] != x[0]:
#     center_yzx = ideal_3D_center(list_mean_points[2],x)
#     radius_yzx = np.linalg.norm(vectorize(center_yzx)-vectorize(x))
#     v = list_mean_points[2]
#     plane_vector_3 = [np.imag(v[0])-np.imag(x[0]),np.real(x[0])-np.real(v
#     [0]),(np.imag(v[0])-np.imag(x[0]))*np.real(v[0]) + (np.real(x[0])-np.real(
#     v[0]))*np.imag(v[0])]
#     bar_2 = intersection_ideal_spheres(vectorize(center_xyz), radius_xyz,
#     vectorize(center_yzx), radius_yzx, plane_vector_1,plane_vector_3)
#     v = list_mean_points[2]
#     plane_vector_4 = [np.imag(v[0])-np.imag(x[0]),np.real(x[0])-np.real(v
#     [0]),(np.imag(v[0])-np.imag(x[0]))*np.real(v[0]) + (np.real(x[0])-np.real(
#     v[0]))*np.imag(v[0])]
#     bar_3 = intersection_ideal_spheres(vectorize(center_xzy), radius_xzy,
#     vectorize(center_yzx), radius_yzx, plane_vector_3,plane_vector_4)
#print(bar_1,bar_2,bar_3)
return list_mean_points, bar_1

#funcion
def simplex_barycenter(x,y,z,t):
    list_xyz = barycenter(x,y,z)
    b_xyz = list_xyz[1]
    v = b_xyz
    plane_vector_1 = [np.imag(v[0])-np.imag(t[0]),np.real(t[0])-np.real(v[0]),(np.
        imag(v[0])-np.imag(t[0]))*np.real(v[0]) + (np.real(t[0])-np.real(v[0]))*np
        .imag(v[0])]

    list_yzt = barycenter(y,z,t)
    b_yzt = list_yzt[1]
    center_xyz = ideal_3D_center(b_xyz, t)
    radius_xyz = np.linalg.norm(vectorize(center_xyz)-vectorize(t))

    center_yzt = ideal_3D_center(b_yzt, x)
    radius_yzt = np.linalg.norm(vectorize(center_yzt)-vectorize(x))
    v = b_yzt
    plane_vector_2 = [np.imag(v[0])-np.imag(x[0]),np.real(x[0])-np.real(v[0]),(np.
        imag(v[0])-np.imag(x[0]))*np.real(v[0]) + (np.real(x[0])-np.real(v[0]))*np
        .imag(v[0])]

    bar = intersection_ideal_spheres(vectorize(center_yzt), radius_yzt, vectorize(
        center_xyz), radius_xyz, plane_vector_1,plane_vector_2)
#print(bar)
return list_xyz[0],list_xyz[1], list_yzt[0],list_yzt[1], bar

```

```

def barycentrical_subdivision(x,y,z): #Calcula la subdivision baricentrica dado un
    triangulo hiperbolico en H3
    list_mean_points, bar = barycenter(x,y,z)
    triangle_list = [[x,bar,list_mean_points[0]], [x,bar,list_mean_points[1]], [y,
        bar,list_mean_points[2]], [y,bar,list_mean_points[0]], [z,bar,
        list_mean_points[1]], [z,bar,list_mean_points[2]]]
    return triangle_list

def Volume_Tetrahedra_from_Vertices(v1,v2,v3,v4):
    l1 = hyperbolic_3D_metric(v1, v2)
    l2 = hyperbolic_3D_metric(v1, v3)
    l3 = hyperbolic_3D_metric(v1, v4)
    l5 = hyperbolic_3D_metric(v2, v4)
    l4 = hyperbolic_3D_metric(v3, v4)
    l6 = hyperbolic_3D_metric(v2, v3)
    Vol = volume_tetrahedra.Volume_Tetrahedron(l1, l2, l3, l4, l5, l6)
    Vol = np.real(Vol)
    return Vol

def hyperbolic_3D_sampling(x,y,z,t,delta):
    token = False
    a=x
    b=y
    c=z
    d=t
    Vol_T = Volume_Tetrahedra_from_Vertices(a, b, c, d)
    counter = 0
    while token == False:
        counter = counter + 1
        #print('Iteracion: ', counter)
        list_abc, bar_abc, list_bcd, bar_bcd, bar = simplex_barycenter(a, b, c, d)
        list_abd, bar_abd = barycenter(a,b,d)
        list_acd, bar_acd = barycenter(a,c,d)
        T_list = [[a,bar_abc, list_abc[0], bar],[a,bar_abc,list_abc[1],bar],[b,
            bar_abc,list_abc[2],bar],[b,bar_abc,list_abc[0],bar],[c,bar_abc,
            list_abc[2],bar],[c,bar_abc,list_abc[1],bar],
            [b,bar_bcd, list_bcd[0], bar],[b,bar_bcd,list_bcd[1],bar],[c,
            bar_bcd,list_bcd[2],bar],[c,bar_bcd,list_bcd[0],bar],[d,
            bar_bcd,list_bcd[2],bar],[d,bar_bcd,list_bcd[1],bar],
            [a,bar_abd, list_abd[0], bar],[a,bar_abd,list_abd[1],bar],[b,
            bar_abd,list_abd[2],bar],[b,bar_abd,list_abd[0],bar],[d,
            bar_abd,list_abd[2],bar],[d,bar_abd,list_abd[1],bar],
            [a,bar_acd, list_acd[0], bar],[a,bar_acd,list_acd[1],bar],[c,
            bar_acd,list_acd[2],bar],[c,bar_acd,list_acd[0],bar],[d,
            bar_acd,list_acd[2],bar],[d,bar_acd,list_acd[1],bar]]

```

---

```

#print(T_list)
V_list = []
for T in T_list:
    V_list.append(Volume_Tetrahedra_from_Vertices(T[0], T[1], T[2],T[3]))
#print(V_list)
P_list = np.array(V_list)/sum(V_list) #Vol_T (con Vol_T no siempre da
    exactamente 1, posible error (?))
p = rd.random()
cumulative = 0
signal = False
index = 0
while signal == False:
    P = P_list[index]
    cumulative = cumulative + P
    if p < cumulative:
        index_T = index
        signal = True
    else:
        index = index + 1
Chosen_T = T_list[index_T]
#print('Subtetrahedron elegido: ', Chosen_T)
Vol_T = V_list[index_T]
if Vol_T < 0:
    #print('Warning!, Volumen: ', Vol_T, ' negativo')
    Vol_T = 0
v1 = Chosen_T[0]
v2 = Chosen_T[1]
v3 = Chosen_T[2]
v4 = Chosen_T[3]
l1 = hyperbolic_3D_metric(v1, v2)
l2 = hyperbolic_3D_metric(v1, v3)
l3 = hyperbolic_3D_metric(v1, v4)
l5 = hyperbolic_3D_metric(v2, v4)
l4 = hyperbolic_3D_metric(v3, v4)
l6 = hyperbolic_3D_metric(v2, v3)
if max(l1, l2, l3, l4, l5, l6) < delta:
    unrelevant_list, unrelevant_bar, unrelevant_list_2, unrelevant_bar_2,
        last_bar = simplex_barycenter(Chosen_T[0], Chosen_T[1], Chosen_T[2],
            Chosen_T[3])
    token = True
else:
    a = Chosen_T[0]
    b = Chosen_T[1]
    c = Chosen_T[2]
    d = Chosen_T[3]
return last_bar

```

**BOOTSTRAP**

```

import numpy as np
import os
import pandas as pd
from os import walk
import matplotlib.pyplot as plt
import random
from itertools import repeat
import collections
from multiprocessing import Pool

manifold_symbol = 'm040(-4,3)'
number_cores = 4

def read_diagrams(manifold_symbol, degree):
    path = 'TU_CARPETA/Bootstrap-H' + str(degree) + '/' + manifold_symbol
    files_list = []
    for (dirpath, dirnames, filenames) in walk(path):
        files_list.extend(filenames)
        break

    List_Diagrams = []
    for string in files_list:
        dataframe = pd.read_csv('TU_CARPETA/Bootstrap-H' + str(degree) + '/' +
            manifold_symbol + '/' + string)
        D = dataframe[['Birth', 'Death']].values.tolist()
        List_Diagrams.append(D)
    return List_Diagrams

def landscape(diagram, k, z): #evaluate k-landscape of the diagram in a point x
    values_list = []
    for p in diagram:
        values_list.append(diagram_point_function(p, z))
    values_array = np.array(values_list)
    ordered_array = np.sort(values_array)
    L = len(values_list)
    if L < k :
        value = 0
    else:
        value = ordered_array[L-k]
    return value

def diagram_point_function(diagram_point, z): #calcula el valor del landscape
    local en el punto z
    x = (diagram_point[1] - diagram_point[0])/2
    y = (diagram_point[1] + diagram_point[0])/2

```

---

```

# b = y-x
# d = y+x
if z <= y-x or z >= y+x:
    value = 0
elif y-x < z and z <= y:
    value = z-y+x
else:
    value = x+y - z
return value

# def affine_function(p0,p1,t):
#     value = (p1[1]-p0[1])*(t-p0[0])/(p1[0]-p0[0]) + p0[1]
#     return value

def plot_landscape(diagram,k,a,b, eps):
    x_array = np.arange(a,b,eps)
    y_array = np.array([landscape(diagram, k, z) for z in x_array])
    plt.plot(x_array, y_array)
    plt.show()
    return None

def mean_landscape(Diagram_List, k, z):
    N = len(Diagram_List)
    value = 0
    for D in Diagram_List:
        value = value + landscape(D, k, z)
    value = value/N
    return value

def bootstrap_difference(Diagram_List, k, z):
    N = len(Diagram_List)
    random_choice = [random.choice(range(N)) for i in range(N)]
    Random_Diagram_List = [Diagram_List[index] for index in random_choice]
    value = abs(mean_landscape(Diagram_List, k, z)-mean_landscape(
        Random_Diagram_List, k, z))
    return value

def bootstrap_quantile_computation(Diagram_List, k, B, eps, alpha): # alpha = 0.95
    N = len(Diagram_List)
    max_list = []
    for D in Diagram_List:
        if len(D) >0 :
            max_list.append(max([p[1] for p in D]))
        else:
            max_list.append(0)
    x_max = max(max_list)

```

```

x_array = np.arange(0,x_max,eps)
mean_array = np.array([mean_landscape(Diagram_List,k,x) for x in x_array])
bootstrap_values = []
for l in range(B):
    b_value = 0
    random_choice = [random.choice(range(N)) for i in range(N)]
    Random_Diagram_List = [Diagram_List[index] for index in random_choice]
    for i in range(len(x_array)):
        b_value = max(b_value, abs(mean_landscape(Random_Diagram_List, k,
            x_array[i])-mean_array[i]))
    bootstrap_values.append(b_value)
bootstrap_array = np.array(bootstrap_values)
sorted_bootstrap = np.sort(bootstrap_array)
index_alpha = int(alpha*B)
q = sorted_bootstrap[index_alpha]
plt.plot(x_array, mean_array)
up_array = np.array([x+q for x in mean_array])
low_array = np.array([max(0,x-q) for x in mean_array])
plt.plot(x_array, up_array, linestyle='dashed')
plt.plot(x_array, low_array, linestyle='dashed')
plt.show()
return q

def process_bootstrap(l,Diagram_List, k, mean_array, x_array):
    print(['Bootstrap:', l])
    N = len(Diagram_List)
    random_choice = [random.choice(range(N)) for i in range(N)]
    Random_Diagram_List = [Diagram_List[index] for index in random_choice]
    b_aux = []
    for i in range(len(x_array)):
        b_aux.append(abs(mean_landscape(Random_Diagram_List, k, x_array[i])-
            mean_array[i]))
    return b_aux

def quantile_showdown(Diagram_List_0,Diagram_List_1, k, eps, q_0, q_1, alpha):
    max_list = []
    for D in Diagram_List_0:
        if len(D) > 0:
            max_list.append(max([p[1] for p in D]))
    for D in Diagram_List_1:
        if len(D) > 0:
            max_list.append(max([p[1] for p in D]))
    x_max = max(max_list)
    x_array = np.arange(0,x_max,eps)
    mean_array_0 = np.array([mean_landscape(Diagram_List_0, k, x) for x in x_array
        ])
    mean_array_1 = np.array([mean_landscape(Diagram_List_1, k, x) for x in x_array

```

```

    ])
    b_value = 0
    for i in range(len(x_array)):
        b_value = max(b_value, abs(mean_array_1[i]-mean_array_0[i]))
    if b_value > q_0+q_1:
        print('Non homeomorphic 3-manifolds with confidence: ', alpha)
    else:
        print('Possibly Homeomorphic 3-manifolds with confidence: ', alpha)
    plt.figure(0)
    plt.plot(x_array, mean_array_1)
    up_array_0 = np.array([x+q_0+q_1 for x in mean_array_0])
    low_array_0 = np.array([max(0,x-q_0-q_1) for x in mean_array_0])
    plt.plot(x_array, up_array_0, linestyle='dashed')
    plt.plot(x_array, low_array_0, linestyle='dashed')
    plt.figure(1)
    plt.plot(x_array, mean_array_0)
    up_array_1 = np.array([x+q_0+q_1 for x in mean_array_1])
    low_array_1 = np.array([max(0,x-q_0-q_1) for x in mean_array_1])
    plt.plot(x_array, up_array_1, linestyle='dashed')
    plt.plot(x_array, low_array_1, linestyle='dashed')
    return None

```

## VOLUMEN DEL TETRAEDRO

```

import numpy as np
from scipy.special import spence

# def conjugate(x): #conjugate of a point in H3, x = (z,h), z complex and h>0
#     x[1] = -x[1]
#     return x

# def vectorize(x): # convert (z,h) to an usual 3D-array
#     vector = np.array([np.real(x[0]),np.imag(x[0]),x[1]])
#     return vector

# def hyperbolic_3D_metric(x,y): #hyperbolic metric in the half space model H3,
#     points are pairs (z,h) with z complex and h real
#     vector_x = np.array([np.real(x[0]),np.imag(x[0]),x[1]])
#     vector_y = np.array([np.real(y[0]),np.imag(y[0]),y[1]])
#     vector_y_conj = np.array([vector_y[0],vector_y[1],-vector_y[2]])
#     a = np.linalg.norm(vector_x-vector_y)/np.linalg.norm(vector_x-vector_y_conj)
#     value = np.log((1+a)/(1-a))
#     return value

#definimos U numerico
def U(a,b,c,d,e,f,z):
    U=spence(1-z)+spence(1-a*b*d*e*z)+spence(1-a*c*d*f*z)+spence(1-b*c*e*f*z)-
        spence(1+a*b*c*z)-spence(1+a*e*f*z)-spence(1+b*d*f*z)-spence(1+c*d*e*z)
    return U

```

```

def zparcialUz(a,b,c,d,e,f,z):
    zparcialUz=-np.log(1-z)-np.log(1-a*b*d*e*z)-np.log(1-a*c*d*f*z)-np.log(1-b*c*e
        *f*z)+np.log(1+a*b*c*z)+np.log(1+a*e*f*z)+np.log(1+b*d*f*z)+np.log(1+c*d*e
        *z)
    return zparcialUz

#formula prop 4.3
def aparcialUa(a,b,c,d,e,f,z):
    aparcial=-np.log(1-a*b*d*e*z)-np.log(1-a*c*d*f*z)+np.log(1+a*b*c*z)+np.log(1+a
        *e*f*z)
    return aparcial

def bparcialUb(a,b,c,d,e,f,z):
    bparcial=-np.log(1-a*b*d*e*z)-np.log(1-b*c*e*f*z)+np.log(1+a*b*c*z)+np.log(1+b
        *d*f*z)
    return bparcial

def cparcialUc(a,b,c,d,e,f,z):
    cparcial=-np.log(1-a*c*d*f*z)-np.log(1-b*c*e*f*z)+np.log(1+a*b*c*z)+np.log(1+c
        *d*e*z)
    return cparcial

def dparcialUd(a,b,c,d,e,f,z):
    dparcial=-np.log(1-a*b*d*e*z)-np.log(1-a*c*d*f*z)+np.log(1+b*d*f*z)+np.log(1+c
        *d*e*z)
    return dparcial

def eparcialUe(a,b,c,d,e,f,z):
    eparcial=-np.log(1-a*b*d*e*z)-np.log(1-b*c*e*f*z)+np.log(1+a*e*f*z)+np.log(1+c
        *d*e*z)
    return eparcial

def fparcialUf(a,b,c,d,e,f,z):
    fparcial=-np.log(1-a*c*d*f*z)-np.log(1-b*c*e*f*z)+np.log(1+a*e*f*z)+np.log(1+b
        *d*f*z)
    return fparcial

def Volume_Tetrahedron(l1,l2,l3,l4,l5,l6):
    G=np.array([[ -1,-np.cosh(l4),-np.cosh(l5),-np.cosh(l3)],[-np.cosh(l4),-1,-np.
        cosh(l6),-np.cosh(l2)],[-np.cosh(l5),-np.cosh(l6),-1,-np.cosh(l1)],[-np.
        cosh(l3),-np.cosh(l2),-np.cosh(l1),-1]])
    detG=complex(np.linalg.det(G))
    a=-np.exp(l4)
    b=-np.exp(l5)
    c=-np.exp(l6)
    d=-np.exp(l1)

```

```

e=-np.exp(l2)
f=-np.exp(l3)
q=np.exp(l1+l4) + np.exp(l2+l5)+np.exp(l3+l6)-np.exp(l1+l2+l3)-np.exp(l1+l5+l6)
-np.exp(l2+l4+l6)-np.exp(l3+l4+l5)+np.exp(l1+l2+l3+l4+l5+l6)
zmenos=2/q*(np.sinh(l1)*np.sinh(l4)+np.sinh(l2)*np.sinh(l5)+np.sinh(l3)*np.
sinh(l6)-np.sqrt(detG))
zmas=2/q*(np.sinh(l1)*np.sinh(l4)+np.sinh(l2)*np.sinh(l5)+np.sinh(l3)*np.sinh(
l6)+np.sqrt(detG))
V=1j/4*(U(a,b,c,d,e,f,zmenos)-U(a,b,c,d,e,f,zmas)-zparcialUz(a,b,c,d,e,f,
zmenos)*np.log(zmenos)+zparcialUz(a,b,c,d,e,f,zmas)*np.log(zmas))
parcialV_11=1j/4*(dparcialUd(a,b,c,d,e,f,zmenos)-dparcialUd(a,b,c,d,e,f,zmas))
parcialV_12=1j/4*(eparcialUe(a,b,c,d,e,f,zmenos)-eparcialUe(a,b,c,d,e,f,zmas))
parcialV_13=1j/4*(fparcialUf(a,b,c,d,e,f,zmenos)-fparcialUf(a,b,c,d,e,f,zmas))
parcialV_14=1j/4*(aparcialUa(a,b,c,d,e,f,zmenos)-aparcialUa(a,b,c,d,e,f,zmas))
parcialV_15=1j/4*(bparcialUb(a,b,c,d,e,f,zmenos)-bparcialUb(a,b,c,d,e,f,zmas))
parcialV_16=1j/4*(cparcialUc(a,b,c,d,e,f,zmenos)-cparcialUc(a,b,c,d,e,f,zmas))
Vol=V-11*parcialV_11-12*parcialV_12-13*parcialV_13-14*parcialV_14-15*
parcialV_15-16*parcialV_16
return Vol

```

## CAMBIOS DE MODELO

```
from pyquaternion import Quaternion
```

```
import numpy as np
import math
import csv
```

```
##Calcula la distancia hiperbolica ente puntos de H3 con modelo disco
```

```
def hyperbolic_distance(p,q):
    v = p - q
    distance = math.acosh(1 + (2*np.linalg.norm(v,axis=0)**2)/((1 - np.linalg.norm
(p,axis=0)**2)*(1 - np.linalg.norm(q,axis=0)**2)))
    return(distance)
```

```
#Del semiespacio al disco (coordenadas cuaternonicas):
```

```
def disk_to_halfspace(p):
    Qp = Quaternion(np.array([p[0],0,p[1],p[2]]))
    imaginary = Quaternion(np.array([0,0,1,0]))
    w = -(Qp-1).inverse*(Qp+1)*imaginary
    return w
```

```
#Del modelo del disco al semiespacio (coordenadas cuaternonicas)
```

```
def halfspace_to_disk(p):
    Qp = Quaternion(np.array([p[0],p[1],p[2],0]))
    imaginary = Quaternion(np.array([0,0,1,0]))
    w = (Qp-imaginary)*(Qp+imaginary).inverse
    return w
```



# Bibliografía

- [1] Culler, M.; Dunfield, N. M.; Goerner, M. y Weeks, J. R. *SnapPy, a computer program for studying the geometry and topology of 3-manifolds*. <http://snappy.computop.org>
- [2] Tralie, C.; Nathaniel Saul, N. y Bar-On, R. (2018). *Ripser.py: A Lean Persistent Homology Library for Python*, The Journal of Open Source Software. <https://doi.org/10.21105/joss.00925>.
- [3] Carlsson, G. (2009). *Topology and Data*, Bulletin of the American Mathematical Society, Volume 46, Number 2, 255–308.
- [4] Cohen-Steiner, D.; Edelsbrunner, H; y Harer, J. (2007). *Stability of Persistence Diagrams*, Discrete Comput Geom 37, 103–120. DOI: 10.1007/s00454-006-1276-5
- [5] Bubenik, P. (2015). *Statistical Topological Data Analysis using Persistence Landscapes*, Journal of Machine Learning Research 16, 77-102.
- [6] Chazal, F.; Fasy, B. T.; Lecci, F.; Rinaldo, A.; Singh, A. y Wasserman, L. (2013). *On the Bootstrap for Persistence Diagrams and Landscapes*, Модел. и анализ информ. систем. Т. 20, №6, 175–185.
- [7] A. Hatcher. (2002). *Algebraic Topology*, Cambridge University Press, Cambridge, 2002. xii+544 pp. ISBN: 0-521-79160-X; 0-521-79540-0.
- [8] Eilenberg, S. (1944). *Singular homology theory*, Ann. of Math. (2) 45, 407–447. MR0010970 (6:96f)
- [9] Dumas, J.G.; Heckenbach, F.; Saunders, B.D. y Welker, V. (2003). *Computing simplicial homology based on efficient Smith normal form algorithms*, In Algebra, Geometry, and Software Systems, 177-207. MR2011758 (2004i:55009)
- [10] A. Björner, A. (1995). *Topological methods*, Handbook of Combinatorics, Vols. 1, 2, 1819–1872, Elsevier, Amsterdam. MR1373690 (96m:52012)

- 
- [11] Carlsson, G. y de Silva, V. (2004). *Topological estimation using witness complexes*, Symposium on Point-Based Graphics, ETH, Zürich, Switzerland.
- [12] Mac Lane, S. (1998). *Categories for the Working Mathematician*, Second edition. Graduate Texts in Mathematics, 5. Springer-Verlag, New York. xii+314 pp. ISBN: 0-387-98403-8. MR1712872 (2001j:18001)
- [13] Zomorodian, A. y Carlsson, G. (2005). *Computing persistent homology*, *Discrete and Computational Geometry*, 33. <https://doi.org/10.1007/s00454-004-1146-y>
- [14] Gabriel, P. (1972). *Unzerlegbare Darstellungen I*. *Manuscripta Mathematica*, 6 : 71–103.
- [15] Milnor, J. (1963) *Morse Theory*. Princeton University Press, New Jersey.
- [16] Edelsbrunner, H.; Letscher, D. y Zomorodian, A. (2002). *Topological persistence and simplification*. *Discrete Comput. Geom.* 28, 511–533.
- [17] Robins, V. (1999). *Toward computing homology from finite approximations*. *Topology Proceedings* 24 , 503–532.
- [18] Chazal, F.;Cohen-Steiner, D.;Glisse, M.;Guibas, L. J. y Oudot, S. Y. (2009). *Proximity of persistence modules and their diagrams* En SCG, 237–246.
- [19] Efron, B. (1979). *Bootstrap methods: Another look at the jackknife*. *Ann. Statist.* , 1–26.
- [20] Kapovich, M. (2000). *Hyperbolic Manifolds and Discrete Groups*. *Progress in Mathematics*. DOI 10.1007/978-0-8176-4913-5
- [21] Bullett, S. (2013). *Holomorphic Dynamics and Hyperbolic Geometry*. Notes for Queen Mary University of London.
- [22] Bourdon, M. (2010). *Mostow type rigidity theorems*. *Handbook of Group Actions (Vol. IV)*.
- [23] Zhang, B. (2014). *On Mostow's Rigidity Theorem*. <https://api.semanticscholar.org/CorpusID:201922409>
- [24] Thurston, W. *The Geometry and Topology of Three Manifolds*. (2002). <http://library.msri.org/nonmsri/gt3m/>
- [25] Vaisala, J. (1971). *Lectures on  $n$ -dimensional Quasiconformal Mappings*. Berlin: Springer-Verlag.
- [26] Ahlfors, L. V. (2006). *Lectures on Quasiconformal Mappings*, 2nd Edition. Providence, Rhode Island: AMS.

- 
- [27] Mostow, G. D. (1968). *Quasi-conformal Mappings in  $n$ -space and the Rigidity of Hyperbolic Space Forms*. Publications Mathématiques De L'IHÈS Vol. 34
- [28] Kornfeld, I. P.; Fomin, S. V. y Sinai, YA G. (1982). *Ergodic Theory*. New York: Springer-Verlag.
- [29] Matoušek, J. (2003). *Using the Borsuk-Ulam Theorem*. Lectures on Topological Methods in Combinatorics and Geometry.
- [30] Rolfsen, D. (1976). *Knots and Links*.
- [31] Murakami, J. y Ushijima, A. (2005). *A volume formula for hyperbolic tetrahedra in terms of edge lengths*. J. Geom. 83 153-163. [arXiv:math/0402087](https://arxiv.org/abs/math/0402087)[math.MG].
- [32] Maskit, B. (1988). *Kleinian Groups*. Springer-Verlag.