



Análisis Topológico de Datos y  
Aprendizaje Automático en la  
Detección De Ríos Atmosféricos

David Vázquez Fernández





# Análisis Topológico de Datos y Aprendizaje Automático en la Detección De Ríos Atmosféricos

David Vázquez Fernández

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

Tutorizada por

Prof. María del Rocío González Díaz



# Índice general

<b>Resumen</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>1. Introducción</b>	<b>5</b>
<b>2. Conceptos previos</b>	<b>9</b>
2.1. Homología persistente en complejos cúbicos . . . . .	9
2.1.1. Complejos cúbicos . . . . .	9
2.1.2. Homología en complejos cúbicos . . . . .	12
2.1.3. Complejos cúbicos filtrados a partir de una matriz de datos ( <i>construcción T</i> ) . . . . .	16
2.1.4. Homología persistente en complejos cúbicos filtrados . . . . .	19
2.2. Imágenes de persistencia . . . . .	22
2.3. Modelos de aprendizaje automático. Clasificación supervisada . . . . .	26
2.3.1. Sobreajuste, división de conjuntos, validación cruzada. . . . .	28
2.3.2. Elección del mejor método de aprendizaje automático con validación cruzada anidada LTO . . . . .	29
2.3.3. Evaluación de la calidad de una clasificación supervisada binaria . . . . .	30

**II ANÁLISIS TOPOLÓGICO DE DATOS Y APRENDIZAJE AUTOMÁTICO EN LA  
DETECCIÓN DE RÍOS ATMOSFÉRICOS**

2.4. Máquina Vector Soporte . . . . .	32
2.4.1. Datos separables linealmente . . . . .	32
2.4.2. Datos cuasi-separables linealmente . . . . .	33
2.4.3. Datos no linealmente separables . . . . .	37
2.5. KNN, método de los K Vecinos más Cercanos . . . . .	38
<b>3. Estado del arte</b>	<b>41</b>
3.1. Datos empleados . . . . .	41
3.2. Esquema del método . . . . .	43
3.3. Primera etapa, análisis topológico de datos . . . . .	44
3.3.1. Gráficos de evolución . . . . .	46
3.3.2. Ejemplo simple del funcionamiento del algoritmo . . . . .	49
3.4. Segunda etapa, clasificación con máquina vector soporte . . . . .	51
3.5. Resultados . . . . .	54
<b>4. Método propuesto</b>	<b>57</b>
4.1. Datos empleados . . . . .	58
4.2. Esquema del método y diferencias con el método existente . . . . .	60
4.3. Primera etapa, análisis topológico de datos . . . . .	61
4.3.1. Obtención de función de filtrado . . . . .	63
4.3.2. Ejemplo simple del funcionamiento del algoritmo . . . . .	66
4.3.3. Generación de vectores resultado. Imágenes de persistencia . . . . .	68
4.3.4. Esquema final y ejemplo de la primera etapa . . . . .	71
4.4. Segunda etapa, método de los K Vecinos más Cercanos . . . . .	73
4.5. Implementación y código . . . . .	77

4.6. Resultados . . . . .	77
4.6.1. Importancia de las dimensiones empleadas en homología y del método de aprendizaje automático escogido . . . . .	78
<b>5. Comparación de métodos: Estabilidad y resultados numéricos</b>	<b>81</b>
5.1. Estabilidad . . . . .	81
5.1.1. Estabilidad de la primera etapa del método propuesto . . . . .	83
5.1.2. Carencia de estabilidad del método de Muszynski, G. et al.	91
5.2. Comparación de resultados numéricos . . . . .	95
<b>6. Conclusiones y trabajo futuro</b>	<b>99</b>
6.1. Conclusiones . . . . .	99
6.2. Trabajo futuro . . . . .	100
<b>Bibliografía</b>	<b>103</b>





# Resumen

Un río atmosférico (RA) es una banda alargada y estrecha de humedad muy concentrada en la parte inferior de la troposfera que va desde los trópicos hasta un continente. La selección de umbrales para definir cuantitativamente un RA continúa siendo un problema abierto. Es por ello que se han hecho esfuerzos en desarrollar métodos de detección automática de RA en datos climáticos que no dependan de umbrales prefijados. En este trabajo, primero, estudiamos un método propuesto que extrae información topológica sobre componentes conexas en imágenes climáticas y luego, emplea una Máquina Vector Soporte para clasificar esa información según se corresponda o no con que haya un RA en la imagen de la que proviene. Después, proponemos una mejora del método anterior, extrayendo información topológica, no solo sobre componentes conexas, sino sobre huecos, y utilizando el método de los K Vecinos más Cercanos para clasificar esa información. Ambos métodos no fijan previamente umbrales. Posteriormente, demostramos que el método propuesto en este trabajo es estable considerando un paso suficientemente pequeño y que el método ya existente no lo es. Finalmente, al comparar los resultados numéricos de los dos métodos, se concluye que nuestro método es el mejor en la mayoría de medidas de bondad de clasificación calculadas.

El código de implementación del método desarrollado en esta memoria está disponible en: <https://github.com/davafe/Metodo-deteccion-rios-atmosfericos>



# Abstract

An atmospheric river (AR) is a long, narrow band of highly concentrated moisture in the lower troposphere that runs from the tropics to a continent. The selection of thresholds to quantitatively define an AR remains an open problem. Thus, efforts have been made to develop automatic AR detection methods in climatic data that do not depend on predetermined thresholds. In this work, first, we study a proposed method that extracts topological information about connected components in climatic images and, after that, uses a Support Vector Machine to classify that information according to whether or not there is an AR in the image from which it comes. Then, we propose an improvement of the previous method, extracting topological information, not only about connected components, but also about holes, and using the K Nearest Neighbors method to classify that information. Both methods do not set thresholds in advance. Subsequently, we show that the method proposed in this work is stable considering a sufficiently small step and that the existing method is not. Finally, when comparing the numerical results of the two methods, we conclude that our method is the best in most of the calculated classification evaluation metrics.

The implementation code of the method developed in this report is available at: <https://github.com/davafe/Metodo-deteccion-rios-atmosfericos>



# 1 | Introducción

Con el cambio climático en marcha, cada vez es más importante estudiar el comportamiento de eventos meteorológicos extremos. En particular, la identificación de estos eventos a partir de una gran cantidad de datos climáticos tiene un gran interés para realizar estadísticas sobre frecuencia, intensidad y localización de estos eventos. La detección de los eventos meteorológicos extremos es útil también para estudiar la calidad de los modelos climáticos que intentan caracterizar estos eventos y su evolución con el cambio climático.

Un río atmosférico (RA) es una banda alargada y estrecha de humedad muy concentrada en la parte inferior de la troposfera que va desde los trópicos hasta un continente. Se ha estimado que los ríos atmosféricos transportan sobre el 90 % de todo el vapor de agua que se transporta desde los trópicos hacia los polos a través de las latitudes medias [1], a pesar de que normalmente abarcan en total menos del 10 % de la circunferencia terrestre en cualquier latitud. En la figura 1.1, se pueden ver dos ejemplos de ríos atmosféricos.

Los RAs están frecuentemente asociados con sucesos de precipitación extrema en latitudes medias. En concreto, están relacionados con fuertes tormentas tanto nevosas como lluviosas en la costa oeste estadounidense [2] y en la costa atlántica europea [3]. Debido a la gran cantidad de agua que puede transportar un único RA, estos suponen un gran peligro para la sociedad porque pueden causar grandes inundaciones [4]. Por otra parte, los RAs son claves en contribuir al manto de nieve y en rellenar embalses, por lo tanto, en mitigar sequías, en áreas como el estado de California en Estados Unidos ([5],[6]). Resumidamente, no todos los RAs causan daños, algunos sólo proporcionan lluvia o nieve beneficiosas y cruciales para el abastecimiento de agua.

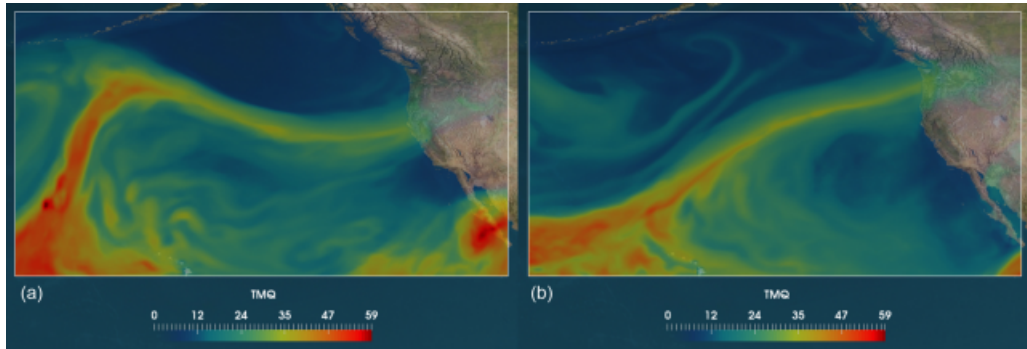


Figura 1.1: Imágenes de ejemplos de ríos atmosféricos alcanzando la costa oeste de EEUU con diferentes trayectorias. Se muestra la densidad superficial de vapor de agua (DSV) ( $kgm^{-2}$ ) de una simulación de la versión 5.1 del Community Atmosphere Model (CAM5.1). Imagen sacada de [7].

El primer reto para la detección de eventos climáticos extremos es construir una definición cuantitativa del evento. Después de ello, se puede desarrollar un algoritmo para identificar y rastrear estos eventos en tiempo y espacio. La definición de RA que hemos dado previamente (banda alargada y estrecha de humedad que va desde los trópicos hacia los polos) es cualitativa y en muchos métodos de detección de RA en datos climáticos se han propuesto definiciones cuantitativas. Estas definiciones cuantitativas tienen que emplear umbrales subjetivos en alguna variable física, por ejemplo, en la densidad superficial de vapor de agua (DSVA) en la columna atmosférica, o en el transporte de densidad lineal de vapor de agua (TDLVA). La selección de buenos umbrales en DSVA o TDLVA continúa siendo un problema abierto. Muchos métodos con los diversos umbrales están recogidos en el *The Atmospheric River Tracking Method Intercomparison Project* ([8]).

Recientemente, se han hecho esfuerzos en desarrollar métodos alternativos de detección de eventos climáticos extremos, como métodos de *deep learning* para reconocimiento de patrones [9], que usan características subyacentes de los conjuntos de datos. En particular, el diseño de estos modelos evita el problema de tener que fijar previamente un umbral cuantitativo.

En la misma línea que el método del *deep learning*, el método desarrollado en [7] emplea reconocimiento de patrones y evita fijar umbrales cuantitativos previos. Este método primero extrae características topológicas de los datos empleando Análisis Topológico de Datos (ATD) y luego, a esas características le

aplica un método de aprendizaje automático, en este caso una Máquina Vector Soporte (MVS), que clasifica los datos como RA o no RA según esas características topológicas. Este método se ha construido específicamente para detectar ríos atmosféricos en el Pacífico Norte, pero podría adaptarse a otras regiones.

Sobre este último procedimiento versará la primera parte nuestro trabajo. Dicho método obtiene un vector de características topológicas de las imágenes en escala de grises obtenidas usando el valor de DSVa a lo largo de la región del estudio. Para ello, modela las imágenes como grafos y estudia la evolución de las componentes conexas de los subgrafos de nivel superior. Estos vectores son los que se clasifican como ríos atmosféricos en la MVS. Como la MVS es un método de clasificación supervisada, para clasificar los vectores necesitamos unas etiquetas que nos digan, para cada vector, si la imagen de la que proviene contiene o no un RA. Con estas etiquetas, entrenamos la MVS, validamos sus hiperparámetros y medimos la calidad de la clasificación.

En la segunda parte de nuestro trabajo, construimos un método de detección que mejora al que acabamos de mencionar. Este método sigue el mismo esquema, pero en la parte del ATD no solo se extrae información topológica de la evolución de las componentes conexas de los subconjuntos de nivel superior, sino que también de los huecos. Para ello, se modelan las imágenes en escala de grises como complejos cúbicos filtrados, se aplica homología persistente en dimensiones 0 y 1 y se obtienen dos diagramas de persistencia, cada uno correspondiente a una de las dos dimensiones consideradas en la homología persistente. Así, el diagrama de persistencia correspondiente a la dimensión 0 aporta información sobre la evolución de las componentes conexas de los subcomplejos de nivel superior, y el diagrama correspondiente a la dimensión 1 aporta información sobre la evolución de los huecos de los subcomplejos de nivel superior. Después, cada diagrama se transforma en una imagen de persistencia, con el fin de obtener vectores para poder clasificarlos posteriormente con un método de aprendizaje automático. Posteriormente, se clasifican estos vectores con un método de clasificación supervisada, el método de los  $K$  Vecinos más Cercanos (*K Nearest Neighbours*, KNN). Otra vez, con la ayuda de las etiquetas, se entrena el KNN, validamos su hiperparámetro y medimos la calidad de la clasificación. Además de medir la calidad de la clasificación del método propuesto, también se estudia la calidad de la clasificación que tendríamos si no empleásemos ambas dimensiones de homología y/o utilizásemos otros métodos de clasificación supervisada.

Podemos destacar dos métodos propuestos recientemente, que, con otros fines, emplean técnicas parecidas a las del método que proponemos nosotros. En [10], se usa homología persistente (en dimensión 1) en complejos cúbicos construidos a partir de una imagen en escala de grises, para estudiar ciclos diarios en la nubosidad de los ciclones tropicales. En [11], se propone un método para clasificar imágenes en escala de grises que, primero, emplea homología persistente en complejos cúbicos y luego aprendizaje automático.

En la última parte de este trabajo, se comparan los dos métodos, el analizado y el propuesto, en términos de estabilidad y resultados numéricos. Se demuestra que el método propuesto en este trabajo es estable considerando un paso suficientemente pequeño y que el método ya existente no lo es. Además, al comparar los resultados numéricos de los dos métodos, se concluye que nuestro método es el mejor en la mayoría de medidas de bondad de clasificación calculadas. Por tanto, el método que proponemos en este trabajo, es mejor en términos de estabilidad y resultados numéricos que el método ya existente.

A continuación, detallamos el contenido de cada capítulo de la memoria. En el capítulo 2 se introducen los conceptos previos necesarios para el trabajo y en el capítulo 3 se describe y analiza el método desarrollado en [7]. Posteriormente, en el capítulo 4 se explica el nuevo método propuesto y en el capítulo 5 se comparan la estabilidad y resultados numéricos de ambos métodos.



## 2 | Conceptos previos

En este capítulo explicamos conceptos previos que se han empleado en este trabajo. En la sección 2.1 se explican los conceptos previos necesarios para calcular la homología persistente en complejos cúbicos, basándonos en [10], [12], [13] y [14]. En la sección 2.2 se definen las imágenes de persistencia y se comenta su utilidad, basándonos en [15]. En la sección 2.3 se explican conceptos sobre modelos de aprendizaje automático, en particular de clasificación supervisada. Nos hemos basado en [17] y [18]. Finalmente, se exponen los métodos de aprendizaje automático que se usan: Máquina Vector Soporte (en la sección 2.4) y los K Vecinos más Cercanos (en la sección 2.5). Para introducir la Máquina Vector Soporte nos hemos basado en [19] y para introducir el concepto de los K Vecinos más Cercanos en [18].

### 2.1 Homología persistente en complejos cúbicos

En la subsección 2.1.1, introducimos los complejos cúbicos y en la subsección 2.1.2 explicamos la homología en complejos cúbicos. En la subsección 2.1.3 explicamos cómo podemos construir complejos cúbicos filtrados a partir de una imagen en escala de grises y, en la subsección 2.1, vemos cómo se puede aplicar homología persistente en esos complejos cúbicos filtrados.

#### 2.1.1 Complejos cúbicos

**| Definición 2.1.** *Un intervalo elemental es un intervalo cerrado  $I \subset \mathbb{R}$  de la forma  $I = [l, l + 1]$  o  $I = [l, l]$  para cierto  $l \in \mathbb{Z}$ . Para simplificar la notación escribimos  $[l] = [l, l]$  en los intervalos que contengan un solo punto.*

Los intervalos elementales que consisten en un solo punto son degenerados, mientras que los de longitud 1 son no degenerados.

**Definición 2.2.** Un cubo elemental  $Q$  es el producto de intervalos elementales, i.e.,  $Q = I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d$  donde  $I_i$  es un intervalo elemental para todo  $i \in \{1, \dots, d\}$ . El conjunto de todos los cubos elementales en  $\mathbb{R}^d$  se denota por  $\mathcal{K}^d$  y el conjunto de todos los cubos se denota por  $\mathcal{K}$ , i.e.,  $\mathcal{K} = \bigcup_{d=1}^{\infty} \mathcal{K}^d$ .

**Definición 2.3.** Sea  $Q = I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d$  un cubo elemental. El número de embebimiento de  $Q$ , denotado por  $\text{emb } Q$ , es la dimensión del espacio euclidiano en el que está contenido, en este caso  $\text{emb } Q = d$  ya que  $Q \subset \mathbb{R}^d$ .

El intervalo  $I_i$  es referido como la componente  $i$ -ésima de  $Q$  y se escribe  $I_i(Q)$ .

La dimensión de  $Q$  se define como el número de componentes no degeneradas de  $Q$  y se denota por  $\text{dim } Q$ .

Observar que si  $\text{emb } Q = d$  entonces  $Q \in \mathcal{K}^d$ . Además, definimos

$$\mathcal{K}_k := \{Q \in \mathcal{K} \mid \text{dim } Q = k\}$$

y

$$\mathcal{K}_k^d := \mathcal{K}_k \cap \mathcal{K}^d$$

para todo  $k, d \in \{0, 1, \dots\}$

**Observación 2.1.** Claramente se tiene que, para cualquier cubo elemental  $Q$ :

$$0 \leq \text{dim } Q \leq \text{emb } Q.$$

En la figura 2.1 podemos ver ejemplos de cubos elementales con distintos números de embebimiento y dimensión.

**Definición 2.4.** Sean  $Q, P \in \mathcal{K}$ .

- $Q$  es una cara de  $P$  si  $Q \subset P$ . Esto se denota como  $Q \preceq P$ .
- $Q$  es una cara propia de  $P$  si  $Q \preceq P$  y  $Q \neq P$ , lo que se denota  $Q \prec P$ .
- $Q$  es una cara primaria si  $Q \subset P$  y  $\text{dim } Q = \text{dim } P - 1$ .

**Ejemplo 2.1.** Sea  $Q = [1, 2] \times [1]$ ,  $R = [1] \times [1]$  y  $P = [1, 2] \times [1, 2]$ . Entonces  $R \preceq Q \preceq P \preceq P$ ,  $R \prec Q \prec P$  y  $Q$  es una cara primaria de  $P$  y  $R$  es una cara primaria de  $Q$ .  $R$  no es una cara primaria de  $P$ .

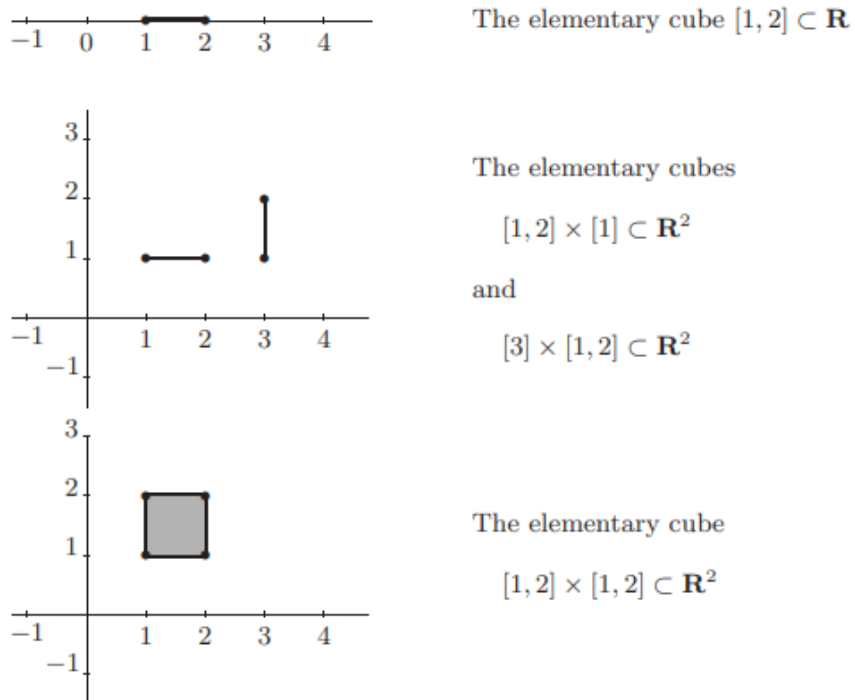


Figura 2.1: Ejemplos gráficos de cubos en  $\mathbb{R}$  y  $\mathbb{R}^2$ . El cubo de arriba tiene número de embebimiento y dimensión 1, los cubos del medio tienen número de embebimiento 2 pero dimensión 1 y, el cubo de abajo tiene dimensión y número de embebimiento 2. Imagen sacada de [12].

**| Definición 2.5.** *Un conjunto  $X \subset \mathbb{R}^d$  es cúbico si  $X$  puede ser escrito como una unión finita de cubos elementales. Dado  $X \subset \mathbb{R}^d$  conjunto cúbico entonces adoptamos la siguiente notación:*

$$\mathcal{K}(X) := \{Q \in \mathcal{K} \mid Q \subset X\}$$

y

$$\mathcal{K}_k(X) := \{Q \in \mathcal{K}(X) \mid \dim Q = k\}.$$

A  $\mathcal{K}(X)$  se le llama *complejo cúbico asociado a  $X$* . Un  $L \subset \mathcal{K}(X)$  es un *subcomplejo cúbico de  $\mathcal{K}(X)$*  si es un *complejo cúbico*, i.e., si  $L = \mathcal{K}(Y)$  para un  $Y \subset X$  que sea *conjunto cúbico*.

De manera análoga a los grafos o a los complejos simpliciales, a los elementos de  $\mathcal{K}_0(X)$  se les llama *vértices* de  $X$ , a los elementos de  $\mathcal{K}_1(X)$  se les llama *aristas* de  $X$  y, de manera general, a los elementos de  $\mathcal{K}_k(X)$  se les llama *k-cubos* de  $X$ .

**Ejemplo 2.2.** Sea  $X = [0, 1] \times [0, 1] \in \mathbb{R}^2$ .  $X$  es un cubo elemental y, por tanto, es un conjunto cúbico. Es fácil de comprobar que:

$$\begin{aligned}\mathcal{K}_2(X) &= \{X\}, \\ \mathcal{K}_1(X) &= \{[0, 1] \times [1], [0, 1] \times [0], [0] \times [0, 1], [1] \times [0, 1]\}, \\ \mathcal{K}_0(X) &= \{[0] \times [0], [0] \times [1], [1] \times [0], [1] \times [1]\}, \\ \mathcal{K}(X) &= \mathcal{K}_2(X) \cup \mathcal{K}_1(X) \cup \mathcal{K}_0(X).\end{aligned}$$

**Observación 2.2.** El análogo a los complejos cúbicos si trabajamos con símlices es el complejo simplicial, donde el conjunto cúbico es lo análogo al espacio subyacente de los complejos simpliciales. Cuando se calcula la homología simplicial, realmente se está calculando la homología en función de la topología del espacio subyacente; de manera análoga, cuando se calcula la homología cúbica, se hace en función de la topología (como subespacio de  $\mathbb{R}^d$  con la topología usual) de los conjuntos cúbicos.

## 2.1.2 Homología en complejos cúbicos

La homología es una herramienta estándar en topología algebraica que nos da un grupo para cada dimensión  $k \in \{0, 1, \dots\}$  y para un espacio topológico  $X$

dado. Cada una de las dimensiones mide distintas propiedades del espacio. En el caso particular de este trabajo, estaremos interesados solamente en homología 0 y 1-dimensional, ya que estamos trabajando con datos bidimensionales (imágenes en escala de grises).

Definamos la homología para un conjunto cúbico. La homología depende del grupo  $G$  que se considere para los coeficientes. En este trabajo vamos a considerar  $G = \mathbb{Z}_2$  y así no vamos a tener en cuenta la orientación. Así, las definiciones que vamos a hacer a continuación asumimos que el grupo que se considera para los coeficientes es  $\mathbb{Z}_2$ .<sup>1</sup>

**| Definición 2.6.** *Sea  $X$  un conjunto cúbico, dado  $k \in \{0, 1, \dots\}$ , una  $k$ -cadena  $c$  con coeficientes en  $\mathbb{Z}_2$  es una combinación lineal formal de  $k$ -cubos en  $X$ , i.e.,*

$$c = \sum_{Q_i \in \mathcal{K}_k(L)} a_i Q_i$$

con coeficientes  $a_i \in \mathbb{Z}_2$  para todo  $Q_i \in \mathcal{K}_k(L)$ .

Podemos sumar estas cadenas de la siguiente forma

$$\sum_{Q_i \in \mathcal{K}_k(X)} a_i Q_i + \sum_{Q_i \in \mathcal{K}_k(X)} b_i Q_i = \sum_{Q_i \in \mathcal{K}_k(X)} (a_i + b_i) Q_i.$$

De esta forma,  $C_k(X) := \{\sum_{Q_i \in \mathcal{K}_k(X)} a_i Q_i \mid a_i \in \mathbb{Z}_2 \forall Q_i \in \mathcal{K}_k(X)\}$  es el conjunto formado por todas las  $k$ -cadenas con coeficientes en  $\mathbb{Z}_2$  formadas por combinación lineal de  $k$ -cubos de  $X$ . Se puede comprobar que  $C_k(X)$  es un grupo abeliano libre con base  $\mathcal{K}_k(X)$ . Dado  $c \in C_k(X)$ , diremos que la cadena  $c$  es de dimensión  $k$ .

Dado  $c = \sum_{Q_i \in \mathcal{K}_k(X)} a_i Q_i \in C_k(X)$ , el soporte de  $c$  es

$$|c| := \cup \{Q_i \in \mathcal{K}_k(X) \mid a_i \neq 0\}.$$

En la definición anterior estamos denotando de igual manera a los  $k$ -cubos que a un objeto algebraico que los identifica.

Una vez tenemos una cadena, una combinación lineal de objetos algebraicos asociados a  $k$ -cubos, obtenemos su soporte, el espacio topológico formado por

---

<sup>1</sup>Si se quiere ver de manera mucho más general la homología cúbica con coeficientes enteros y con resultados y demostraciones, consultar [12]. La adaptación al caso de  $\mathbb{Z}_2$  está basado en la adaptación hecha en [10].

la unión de los  $k$ -cubos cuyos objetos algebraicos asociados tienen coeficiente no nulo en la cadena.

Como los coeficientes de las cadenas pertenecen a  $\mathbb{Z}_2$ , solo pueden valer 0 o 1, esto conlleva a que no hay dos cadenas con el mismo soporte, i.e., a la hora de combinar algebraicamente cubos solo estamos teniendo en cuenta los cubos que combinamos y por lo tanto no tenemos en cuenta orientación, multiplicidad, etc. de los cubos.

*Ejemplo 2.3.* Consideremos  $X = [0, 1] \times [0, 1]$ , un conjunto cúbico. Hemos visto en el ejemplo 2.2 que las aristas de  $X$  son

$$\mathcal{K}_1(X) = \{A_1 := [0, 1] \times [1], A_2 := [0, 1] \times [0], A_3 := [0] \times [0, 1], A_4 := [1] \times [0, 1]\}.$$

Un ejemplo de cadena 1 dimensional  $c \in C_1(X)$  sería:  $c = A_1 + A_2 + A_3 + A_4$ , dónde  $|c|$  es el borde del cuadrado  $X$ .  $X$  (como objeto algebraico) no deja de ser una cadena 2 dimensional.

En la siguiente definición definimos una aplicación que a cada cadena le asigna una cadena “borde”, extendiendo los bordes topológicos de los soportes de las cadenas a las propias cadenas.

**Definición 2.7.** Dado  $k \in \{1, 2, \dots\}$  y un conjunto cúbico  $X$ , definimos el  $k$ -operador borde para el conjunto cúbico  $X$ :

$$\delta_k: C_k(X) \longrightarrow C_{k-1}(X)$$

Es un homomorfismo de grupos abelianos libres, que se define para los elementos  $Q \in \mathcal{K}_k(X)$  y se extiende por linealidad al resto de  $C_k(X)$ . Dado  $Q \in \mathcal{K}_k(X)$ , definimos el borde de  $Q$  como la suma de sus caras primarias

$$\delta_k(Q) := \sum_{\substack{P \prec Q, \\ \dim P = \dim Q - 1}} P \quad .$$

Por tanto, dado  $c \in C_k(X)$ ,  $\delta_k(c) = \sum_{Q_i \in \mathcal{K}_k(X)} a_i \delta_k(Q_i)$ .

Esta última definición está particularizada para el caso de considerar coeficientes en  $\mathbb{Z}_2$ .

Con esta proposición se ve, de manera algebraica, que el borde de un borde de una cadena es 0.

**Proposición 2.1.** Sea  $X$  un conjunto cúbico y  $k \in \{1, 2, \dots\}$ , entonces

$$\delta_{k-1} \circ \delta_k = 0.$$

Podemos encontrar una demostración de esta proposición (en el caso más general de coeficientes enteros) en [12].

Ahora ya estamos preparados para definir los grupos de homología.

**Definición 2.8.** Sea  $X$  un conjunto cúbico y  $k \in \{0, 1, \dots\}$ , el grupo de homología de dimensión  $k$  de  $X$  es:

$$H_k(X) := \text{Ker}(\delta_k) / \text{Im}(\delta_{k+1}).$$

Definimos el grupo de los  $k$ -ciclos como  $Z_k(X) := \text{Ker}(\delta_k)$  y el grupo de los  $k$ -bordes como  $B_k(X) := \text{Im}(\delta_{k+1})$ .

Observemos que los grupos de homología están bien definidos, ya que, por la proposición 2.1, tenemos que  $\text{Im}(\delta_{k+1}) \subset \text{Ker}(\delta_k)$ .

Denotaremos indistintamente a  $H_k(X)$ , como  $H_k(X)$  (grupo de homología de dimensión  $k$  del conjunto cúbico  $X$ ) y como  $H_k(\mathcal{K}(X))$  (grupo de homología de dimensión  $k$  del complejo cúbico  $\mathcal{K}(X)$ ). Esto está en línea con lo comentado en la observación 2.2.

Un  $k$ -ciclo de  $X$  es un elemento cuyo borde es nulo y un  $k$ -borde de  $X$  es un elemento que es borde de una cadena  $k + 1$  dimensional. Así, las clases no nulas de  $H_k(X)$  tienen como representantes  $k$ -ciclos que no son  $k$ -bordes.

Con el siguiente teorema, demostrado en [12], vemos qué información nos aporta  $H_0$ .

**Teorema 2.1.** Sea  $X$  un conjunto cúbico, entonces  $H_0(X)$  es un grupo abeliano libre. Además, si  $\{P_i \in \mathcal{K}_0(X) | i \in \{1, \dots, n\}\}$  es una colección de vértices en  $X$  tal que hay un vértice por cada componente conexa de  $X$ , entonces:

$$\{[P_i] \in H_0(X) | i \in \{1, \dots, n\}\}$$

es una base de  $H_0(X)$ .  $[P_i]$  es la clase de  $H_0(X)$  en la que está  $P_i$

Con este teorema podemos interpretar que existe un elemento de la base de  $H_0(X)$  por cada componente conexa de  $X$ , así que  $H_0(X)$  nos da información sobre el número y cuales son las componentes conexas de  $X$ .

La interpretación que se le puede dar a  $H_1(X)$  es similar: el núcleo de  $\delta_1$ ,  $Z_1(X)$ , está generado por 1-cadenas cuyo borde sea nulo pero no sean borde de una 2-cadena. Es decir,  $H_1(X)$  está generado por cadenas que representan uniones de aristas cuyo borde sea nulo (i.e., estas uniones tienen los vértices repetidos un número par de veces, por lo tanto es una unión de aristas que es cerrada) y que no sean borde de 2-cadenas (i.e., que no sean bordes de una unión de 2-cubos, por lo tanto que esa unión de aristas cerrada no esté rellena por 2-cubos). Por tanto un elemento no nulo de  $H_1(X)$  es una combinación de huecos en  $X$  y de la misma manera, hay un elemento de la base de  $H_1(X)$  por cada hueco en  $X$ .

### 2.1.3 Complejos cúbicos filtrados a partir de una matriz de datos (*construcción $T$* )

En esta subsección veremos lo que es un complejo cúbico filtrado y como podemos construir un complejo cúbico a partir de una matriz de datos. Una matriz de datos se puede ver como una imagen digital en escala de grises de dimensión 2, dónde el “color” de cada píxel (no deja de ser un número) está determinado por el valor de los elementos de la matriz. Para ello veremos cómo se construyen de manera general complejos cúbicos filtrados a partir de una imagen en escala de píxeles de cualquier dimensión.

Con esta construcción podremos aplicar la homología persistente (que explicaremos en la subsección 2.1.4) a una matriz de datos para captar los patrones topológicos que hay en esa matriz de datos.

Empezamos viendo lo que es un complejo cúbico filtrado.

**| Definición 2.9.** Sea  $X \subset \mathbb{R}^q$  un conjunto cúbico, una función  $f: \mathcal{K}(X) \rightarrow \mathbb{R}$  es monótona si  $P \preceq Q$  implica  $f(P) \leq f(Q)$ .

**| Definición 2.10.** Un complejo cúbico filtrado es un par  $(\mathcal{K}(X), f)$  formado por un complejo cúbico asociado a un conjunto cúbico  $X \subset \mathbb{R}^q$ ,  $\mathcal{K}(X)$ , y una función monótona  $f: \mathcal{K}(X) \rightarrow \mathbb{R}$ . A  $f$  se le llama función de filtrado.

**Observación 2.3.** Observar que la condición de monotonía implica que para todo  $r \in \mathbb{R}$ ,  $f^{-1}(-\infty, r]$  incluya a todos los cubos de dimensión  $q$  cuya imagen por  $f$  es menor que  $r$  y a todas las caras propias de estos cubos (ya que su imagen tiene que ser menor o igual a la de los cubos por monotonía). Esto implica que



$f^{-1}(-\infty, r]$  es un subcomplejo cúbico, en particular,

$$f^{-1}(-\infty, r] = \mathcal{K} \left( \bigcup_{k=0}^q \bigcup_{\substack{P \in \mathcal{K}_k(X) \\ f(P) \leq r}} P \right).$$

Veamos qué es una imagen dimensional en escala de grises.

**| Definición 2.11.** *Una imagen (digital)  $d$ -dimensional en escala de grises de tamaño  $(n_1, n_2, \dots, n_d)$  es un array con valores en  $\mathbb{R}$ , i.e.,  $\mathcal{I} \in M_{n_1 n_2 \dots n_d}(\mathbb{R})$ . De manera equivalente, es una función definida en una rejilla:*

$$\mathcal{I}: I = \llbracket 1, n_1 \rrbracket \times \llbracket 1, n_2 \rrbracket \times \dots \times \llbracket 1, n_d \rrbracket \longrightarrow \mathbb{R}$$

donde, dado  $i \in \{1, \dots, d\}$ ,

$$\llbracket 1, n_i \rrbracket := \{k \in \mathbb{N} \mid k \in [1, n_i]\}.$$

El conjunto de índices  $I$  se llama el dominio de la imagen  $\mathcal{I}$ .

Si  $d = 2$ , los elementos de  $I$  se les llama píxeles y, si  $d \geq 3$ , se les llama vóxeles.

Es claro que una matriz  $n \times m$  de datos es una imagen 2-dimensional en escala de grises de tamaño  $(n, m)$ .

Para inducir una topología que represente una imagen en escala de grises y represente la conexión entre sus vóxeles/píxeles no podemos quedarnos con la topología relativa en  $I \subset \mathbb{Z}^d \subset \mathbb{R}^d$ , ya que con esa topología la imagen es totalmente desconexa. Es por ello que vamos a inducir la topología en la imagen a través de complejos cúbicos filtrados.

Existen dos formas de construir un complejo cúbico filtrado a partir de una imagen  $d$  dimensional. Una es la *construcción  $V$*  donde cada vóxel/píxel representa un vértice del complejo cúbico y la función se define en los vértices según el valor del píxel correspondiente y en los cubos de dimensión superior como el máximo de la función en los vértices que son cara de esos cubos. Sin embargo, en este trabajo vamos a usar la *construcción  $T$* , donde cada vóxel/píxel representa un  $d$ -cubo del complejo cúbico y la función se define en los  $d$ -cubos según el valor del píxel correspondiente y en los cubos de dimensión inferior a  $d$  como el mínimo de la función en los  $d$ -cubos que los tienen como cara propia. Podemos ver un ejemplo de comparativo de las dos construcciones en la figura 2.2

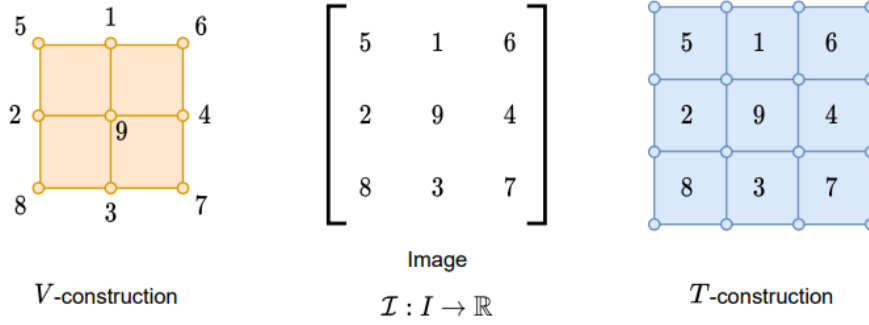


Figura 2.2: Comparación de una construcción  $V$  (izquierda, solo están los valores de la función en los vértices) y una construcción  $T$  (derecha, solo están los valores de la función en los 2-cubos) a partir de la misma matriz (centro). En la construcción  $V$  asociamos un vértice a cada elemento de la matriz y en la  $T$  asociamos un 2-cubo a cada elemento de la matriz. Imagen sacada de [13].

**Definición 2.12.** Dada una imagen  $d$ -dimensional en escala de grises  $\mathcal{I} : I \rightarrow \mathbb{R}$  de tamaño  $(n_1, n_2, \dots, n_d)$ , la construcción  $T$  es un complejo cúbico filtrado  $(T(I), T(\mathcal{I}))$  definido de la siguiente forma:

1.  $T(I) := \mathcal{K}([0, n_1] \times [0, n_2] \times \dots [0, n_d])$ .
2.  $T(\mathcal{I}) : T(I) \rightarrow \mathbb{R}$  definida de la siguiente forma:
  - Si  $P \in \mathcal{K}_d([0, n_1] \times [0, n_2] \times \dots [0, n_d])$  por lo cual

$$P = [i_1, i_1 + 1] \times [i_2, i_2 + 1] \times \dots \times [i_d, i_d + 1]$$

para ciertos  $i_1 \in \{0, \dots, n_1 - 1\}$ ,  $i_2 \in \{0, \dots, n_2 - 1\}$ ,  $\dots$ ,  $i_d \in \{0, \dots, n_d - 1\}$ , entonces definimos:

$$T(\mathcal{I})(P) = \mathcal{I}(i_1 + 1, i_2 + 1, \dots, i_d + 1).$$

- Si  $P \in \mathcal{K}_l([0, n_1] \times [0, n_2] \times \dots [0, n_d])$  con  $l \in \{0, \dots, d - 1\}$ , entonces definimos:

$$T(\mathcal{I})(P) = \text{mín} \left\{ T(\mathcal{I})(Q) \mid Q \in \mathcal{K}_d([0, n_1] \times [0, n_2] \times \dots [0, n_d]), P \prec Q \right\}.$$

Claramente,  $T(\mathcal{I})$  es monótona. Se puede ver un ejemplo simple de una construcción  $T$  a partir de una matriz  $2 \times 2$  en la figura 2.3

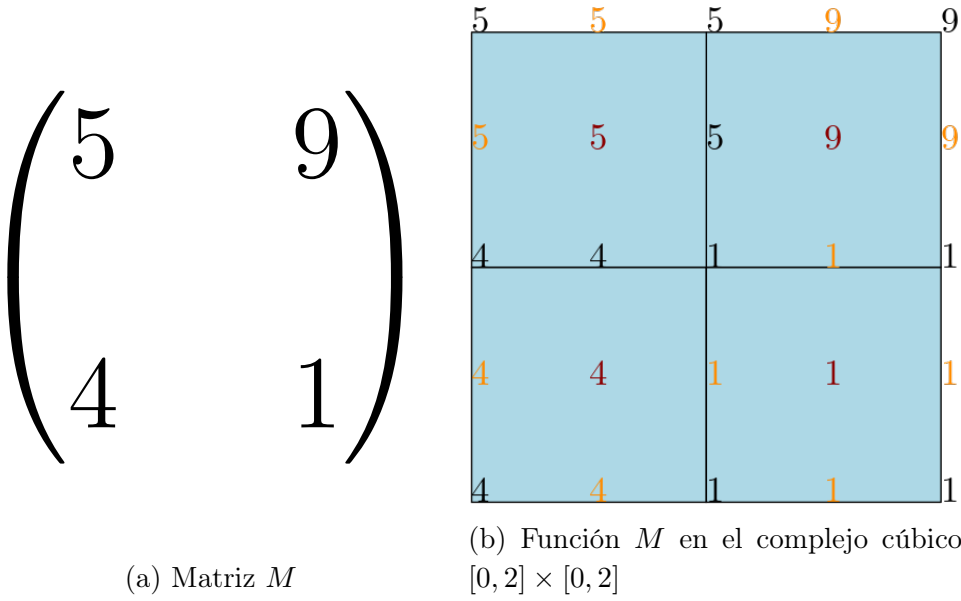


Figura 2.3: Podemos ver una matriz ejemplo en la imagen de la izquierda y en la imagen de la derecha la construcción  $T$  asociada (i.e. un complejo cúbico con una función definida en él).

2.1.4 Homología persistente en complejos cúbicos filtrados

Para un conjunto cúbico fijo  $X$ , los  $H_k(X)$  miden información topológica de ese espacio. La homología persistente tiene como *input* un espacio topológico cambiante y resume cómo va cambiando la homología según cambiemos el espacio. Estos espacios cambiantes, van a ser subcomplejos de nivel inferior asociados a una función de filtrado monótona, definida en el complejo cúbico, para umbrales cambiantes. Estos espacios cambiantes forman lo que llamamos una filtración de un complejo cúbico filtrado, que definimos a continuación:

**Definición 2.13.** Sea  $(\mathcal{K}(X), f)$  un complejo cúbico filtrado. Dado un conjunto de umbrales  $\{a_1, \dots, a_{n-1}\}$ ,  $a_1 < a_2 < \dots < a_{n-1}$  una filtración de  $(\mathcal{K}(X), f)$  es la secuencia de complejos cúbicos

$$\emptyset = K_0 \subset K_1 \subset K_2 \subset \dots \subset K_n = \mathcal{K}(X)$$

donde, para cada  $i \in \{1, 2, \dots, n-1\}$ ,  $K_i := f^{-1}((-\infty, a_i])$  es un complejo cúbico.

Podemos ver un ejemplo de filtración de un complejo cúbico filtrado en la figura 2.4.

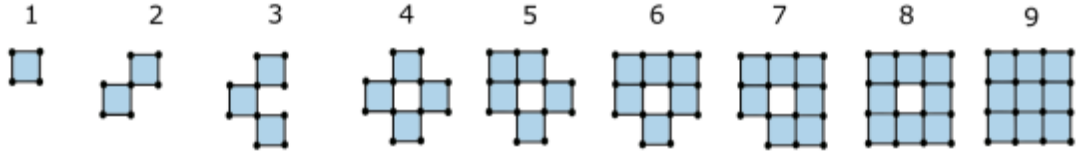


Figura 2.4: Filtración asociada al complejo cúbico construcción  $T$  de la figura 2.2 empleando los umbrales  $1 < 2 < \dots < 9$ . Imagen sacada de [13].

Dada una filtración  $\emptyset = K_0 \subset K_1 \subset K_2 \subset \dots \subset K_n = \mathcal{K}(X)$ , para cada  $i \leq j$ , la inclusión  $K_i \hookrightarrow K_j$  induce un homomorfismo en los grupos de homología  $f_k^{i,j}: H_k(K_i) \rightarrow H_k(K_j)$ . Por tanto, esta filtración se corresponde con una secuencia de grupos de homología conectados por homomorfismos:

$$0 = H_k(K_0) \rightarrow H_k(K_1) \rightarrow H_k(K_2) \rightarrow \dots \rightarrow H_k(K_n) = H_k(X)$$

para cada dimensión  $k \in \{0, 1, \dots, \}$ .

Si vamos de  $K_{i-1}$  a  $K_i$  habrá nuevas clases que aparezcan en  $H_k(K_i)$  que no estaban en  $H_k(K_{i-1})$ , y viceversa, habrá clases que estaban en  $H_k(K_{i-1})$  y que en  $H_k(K_i)$  se han sumergido en otras o se han vuelto triviales. Para rastrear esta evolución de las clases se definen los grupos de homología persistentes.

**Definición 2.14.** Sea  $\emptyset = K_0 \subset K_1 \subset K_2 \subset \dots \subset K_n = \mathcal{K}(X)$  una filtración. Los grupos de homología persistente de dimensión  $k \in \{0, 1, \dots\}$  son  $H_k^{i,j} := \text{Im}(f_k^{i,j})$  para  $0 \leq i \leq j \leq n$ , i.e., son las imágenes de los homomorfismos inducidos por las inclusiones.

Observar que  $H_k^{i,i} = H_k(K_i)$ .  $H_k^{i,j}$  consiste en las clases de homología de  $H_k(K_i)$  que no se han sumergido en otras o se han vuelto triviales en  $H_k(K_j)$ , de manera más formal:

$$H_k^{i,j} = \frac{Z_k(K_i)}{B_k(K_j) \cap Z_k(K_i)}.$$

**Definición 2.15.** Sean  $\emptyset = K_0 \subset K_1 \subset K_2 \subset \dots \subset K_n = \mathcal{K}(X)$  una filtración,  $k \in \{0, 1, \dots\}$ ,  $i, j \in \{1, 2, \dots, n-1\}$  tal que  $i \leq j$  y  $\gamma \in H_k(K_i)$  una clase.

- Se dice que  $\gamma$  nació en  $K_i$  si  $\gamma \notin H_k^{i-1,i}$ , i.e., si no es imagen de ninguna clase por la aplicación  $f_k^{i,j}$ .

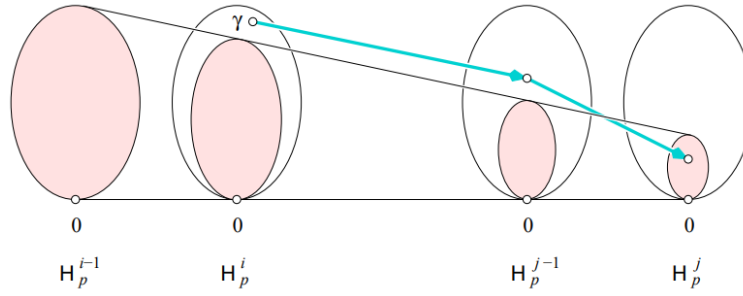


Figura 2.5: Representación gráfica del nacimiento y muerte de una clase  $\gamma$ . La zona rosa de cada grupo de homología se corresponde con la imagen del primer grupo de homología representado (denotado por  $H_p^{i-1}$ ) a través de los homomorfismos inducidos por las inclusiones. La clase  $\gamma$  nace en  $K_i$ , ya que no está (por primera vez) en la imagen de  $f_p^{i-1,i}$  y muere entrando en  $K_j$  ya que es la primera vez que su imagen cae en la imagen de  $H_p^{i-1}$  a través de  $f_p^{i-1,j}$ . Imagen sacada de [14]

- Si  $\gamma$  nació en  $K_i$ , se dice que  $\gamma$  murió entrando en  $K_j$  si se anula en  $H_k(K_j)$  o se une con una clase que proviene de un nivel anterior al movernos de  $K_{j-1}$  hasta  $K_j$ . Esto es, de manera formal, que  $f_p^{i,j-1}(\gamma) \notin H^{i-1,j-1}$  pero que  $f_p^{i,j}(\gamma) \in H^{i-1,j}$ .
- Si  $\gamma$  nace en  $K_i$  y muere entrando en  $K_j$  decimos que tiene persistencia  $a_j - a_i$  y la denotamos por  $\text{pers}(\gamma)$ , donde  $a_j$  y  $a_i$  son los umbrales que se usaron para definir  $K_i$  y  $K_j$  respectivamente. Si  $\gamma$  nace en  $K_i$  y nunca muere, decimos que tiene persistencia infinita, por tanto  $\text{pers}(\gamma) = \infty$

Observar que en la definición estamos diciendo que cuando dos clases se unen, la que se muere es la que ha nacido más tarde (la más “joven”). Esta norma que se emplea es la llamada *Elder Rule*.

En la imagen 2.5 podemos ver gráficamente la definición de nacimiento y muerte de una clase.

Vamos a representar el nacimiento y muerte de esas clases de equivalencia dibujando puntos en un plano. Alguno de esos puntos puede tener coordenada infinita, así que vamos a dibujar en el plano extendido  $\mathbb{R}^2$ .

**| Definición 2.16.** Sea una filtración  $\emptyset = K_0 \subset K_1 \subset K_2 \subset \dots \subset K_n = \mathcal{K}(X)$ ,  $k \in \{0, 1, \dots\}$  y  $i, j \in 1, 2, \dots, n$  tal que  $i < j$ , definimos la multiplicidad de un

punto del diagrama de persistencia de dimensión  $k$   $(a_i, a_j) \in \bar{\mathbb{R}}^2$  (siendo  $a_i, a_j$  los umbrales con los que se definieron respectivamente  $K_i, K_j$ ) como el número de clases independientes que nacen en  $K_i$  y mueren en  $K_j$ . Denotamos esta multiplicidad por  $u_k^{i,j}$ . Para los puntos  $(a_i, \infty) \in \bar{\mathbb{R}}^2$  (siendo  $a_i$  el umbral con el que se definió  $K_i$ ) se define su multiplicidad en el diagrama,  $\mu_k^{i,\infty}$ , como el número de clases independientes que nacen en  $K_i$  y no mueren.

Si dibujamos cada punto de  $\bar{\mathbb{R}}^p$  que tiene multiplicidad positiva (e indicamos cual es) obtenemos el *diagrama de persistencia de dimensión  $p$*  de la filtración. Cada punto del diagrama representa una (o varias si hay más multiplicidad) clase(s) de persistencia(s) cuya persistencia es la distancia del punto a la diagonal (recta  $x = y$ ). Como las multiplicidades están definidas para  $i < j$  todos los puntos están encima de la diagonal. Podemos ver un ejemplo de diagrama de persistencia en la figura 2.6.

Las clases de homología que duran más en a lo largo de una filtración, las más persistentes, se considera que representan unas características topológicas inherentes a los datos. Sin embargo las clases que mueren poco después de nacer, se considera que no representan una característica topológica inherente a los datos sino que representan ruido. Así, cuanto más alejado esté un punto de la diagonal más importante es a la hora de extraer información topológica.

Destacamos que el programa que hemos escogido para calcular la homología persistente de un complejo cúbico filtrado, devuelve el diagrama de persistencia de forma que no va a haber puntos del diagrama con multiplicidad mayor que 1 salvo que haya varias clases que nazcan en el mismo umbral.

## 2.2 Imágenes de persistencia

En esta sección vamos a describir una forma de vectorizar un diagrama de persistencia (descrito en la subsección 2.1.4). Es decir, con las imágenes de persistencia vamos a poder resumir la información de un diagrama de persistencia en un vector. Las imágenes de persistencia son una forma de medir cuántos puntos (de manera ponderada) hay en cada zona del diagrama.

Antes de definir lo que es una imagen de persistencia, vamos a definir lo que es una superficie de persistencia.

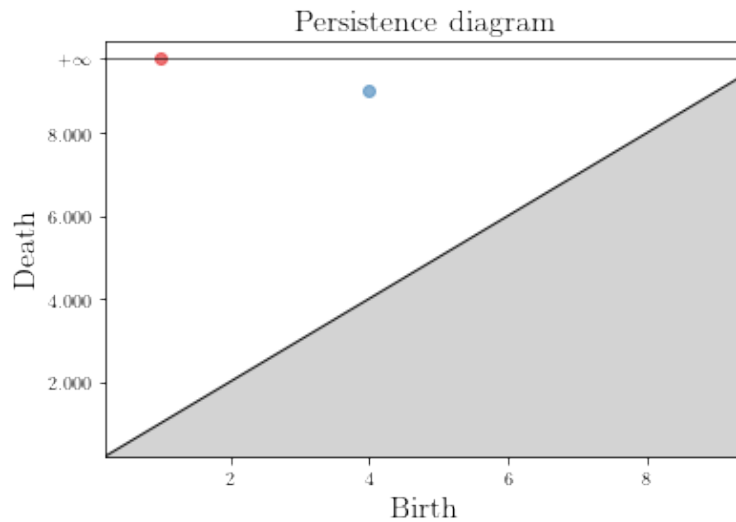


Figura 2.6: Podemos ver dos diagramas de persistencia, correspondientes a la filtración del ejemplo 2.4, uno encima del otro. Los puntos en rojo representan dimensión 0 y los puntos en azul representan la dimensión 1. Respecto de la dimensión 0, podemos ver que solo tenemos un punto, que representa una clase que nace en 1 y no muere. Esta se corresponde con la única componente conexa que aparece en la filtración, que nace en 1 y se mantiene. Respecto de la dimensión 1, podemos ver que solo tenemos un punto, que representa una clase que nace en 4 y muere en 9. Esta se corresponde con el único hueco que aparece en la filtración, que nace en 4 y se muere rellenado en 9.

**| Definición 2.17.** *Sea  $B$  el conjunto formado por todos los puntos de un diagrama de persistencia que tienen multiplicidad positiva y no tienen persistencia infinita, sea*

$$g: \mathbb{R}^2 \longrightarrow \mathbb{R}$$

*una función diferenciable a trozos tal que  $g(x, 0) = 0$  para todo  $x \in \mathbb{R}$ , sea*

$$\phi_u: \mathbb{R}^2 \longrightarrow \mathbb{R}$$

*una función de densidad de probabilidad con media un vector  $u \in \mathbb{R}^2$  dado y sea*

$$\begin{aligned} T: \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ (x, y) &\longmapsto (x, y - x). \end{aligned}$$

*Con todos estos ingredientes, podemos definir la superficie de persistencia de  $B$  como la función:*

$$\begin{aligned} \rho_B: \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ z &\longmapsto \rho_B(z) = \sum_{u \in T(B)} g(u) \phi_u(z). \end{aligned}$$

Dado el conjunto de puntos de un diagrama de persistencia  $B$  (sin los puntos que tienen segunda coordenada infinita), aplicamos la transformación lineal  $T$  para pasar de coordenadas nacimiento-muerte en  $B$  a coordenadas nacimiento-persistencia en  $T(B)$ . Después, obtenemos la superficie de persistencia como la suma ponderada de funciones de densidad centradas en los puntos.

La superficie de persistencia  $\rho_B$  se reduce a un vector finito, fijando una rejilla a lo largo de una región de interés acotada del plano (por ejemplo, la región más pequeña que contiene todos los puntos de  $T(B)$ ) y obtenemos una imagen de persistencia integrando  $\rho_B$  en todos los cubos de la rejilla.

**| Definición 2.18.** *Sea  $B$  el conjunto formado por todos los puntos de un diagrama de persistencia que tienen multiplicidad positiva y no tienen persistencia infinita y  $\rho_B$  una imagen de persistencia del diagrama. Sea  $\{I_1, I_2, \dots, I_n\}$  unos píxeles que cubren como rejilla una región de interés acotada del plano. Así la imagen de persistencia de  $B$  es el vector*

$$\left( I(\rho_B)_{I_1} = \int \int_{I_1} \rho_B dy dx, I(\rho_B)_{I_2} = \int \int_{I_2} \rho_B dy dx, \dots, I(\rho_B)_{I_n} = \int \int_{I_n} \rho_B dy dx \right).$$



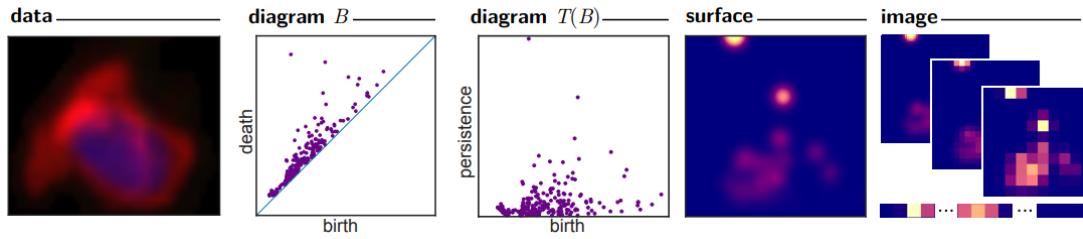


Figura 2.7: Podemos ver el proceso de obtención de una imagen de persistencia a partir de una imagen en escala de grises. En la imagen de la izquierda vemos una imagen en escala de grises como datos de *input*, a su izquierda vemos un diagrama de persistencia  $B$  asociado de una dimensión  $k \in \{0, 1\}$  (sin los puntos que tienen persistencia infinita) y a la izquierda de  $B$  vemos  $T(B)$ , i.e., el diagrama  $B$  cambiando de coordenadas nacimiento-muerte a nacimiento-persistencia. La cuarta imagen por la izquierda es la superficie de persistencia asociada a  $B$ . En la quinta imagen por la izquierda podemos ver varias imágenes de persistencia diferenciadas por la precisión que se usa para integrar la superficie de persistencia anterior. Además, en la quinta foto por la izquierda podemos ver una imagen de persistencia aplanada, formando un vector. Imagen sacada de [15].

En la figura 2.7 podemos ver de manera gráfica la obtención de una imagen de persistencia a partir de una imagen en escala de grises.

Las imágenes de persistencia dependen de 3 parámetros:

- **Resolución de la imagen:** Mide el tamaño de los píxeles considerados al integrar la superficie de persistencia. Cuánto más resolución, menos tamaño de los píxeles y más dimensiones va a tener la imagen de persistencia. Simulaciones en [15], muestran que en métodos de clasificación a partir de imágenes de persistencia el porcentaje de acierto parece robusto frente la elección de resolución. Esto indica que, aparentemente, reducir la dimensionalidad de las imágenes de persistencia al reducir la resolución no afecta mucho a la calidad de la información extraída por las imágenes de persistencia.
- **La distribución  $\phi_u$ .** Normalmente, se suele emplear como  $\phi_u$  una distri-

bución normal centrada en  $u = (u_x, u_y) \in T(B)$  y con matriz de varianzas-covarianzas con  $\sigma^2$  en los elementos de la diagonal y 0 en el resto, i.e., se suele considerar

$$\phi_u = \frac{1}{2\pi\sigma^2} e^{-[(x-u_x)^2+(y-u_y)^2]/2\sigma^2}$$

para un  $u \in T(B)$ . Si se escoge la distribución normal, queda por escoger otro parámetro,  $\sigma$ , la desviación típica de cada variable. Experimentos en [15], muestran que en métodos de clasificación a partir de imágenes de persistencia el porcentaje de acierto es poco sensible frente la elección de  $\sigma$ .

- **Función de ponderación  $g$ .** Para asegurar la estabilidad (ver 5.1), se exige que  $g$  sea continua, diferenciable a trozos y que  $g(x, 0) = 0$  para todo  $x \in \mathbb{R}$ . Estas funciones, además, buscan ponderar más a los puntos con más persistencia, los más alejados de la diagonal en los diagramas, ya que son los que más información topológica sobre la estructura de los datos nos aporta (y es menos probable que sea ruido topológico).

Una opción simple es que  $g$  dependa solo de la segunda coordenada (la de persistencia) y que, para ponderar los puntos del diagrama más cuanto más persistencia tengan se puede considerar una función no decreciente respecto de la segunda coordenada. En connivencia con lo anterior se puede usar  $g(x, y) = y$  o  $g$  una función lineal a trozos en la variable  $y$  que valga 0 si  $y = 0$  y 1 para el punto del diagrama con más persistencia.

## 2.3 Modelos de aprendizaje automático. Clasificación supervisada

Un modelo de aprendizaje automático es un modelo en que una máquina (un ordenador normalmente) obtiene un conocimiento a partir de unos datos que se le aportan y el desempeño de la máquina mejora cuanto más datos se le aportan. "En el aprendizaje automático, un computador observa datos, construye un modelo basado en esos datos y utiliza ese modelo a la vez como una hipótesis acerca del mundo y una pieza de software que puede resolver problemas" ([16, p. 651]).

Podemos clasificar el aprendizaje automático según los datos "a priori" que tenemos.

- El *aprendizaje supervisado* es un tipo de aprendizaje automático en que en

los datos existen una o más variables a predecir (variables dependientes) en función de un conjunto de variables predictoras (variables independientes). Por tanto una manera de evaluar la calidad de los modelos de aprendizaje supervisado es comparar los valores reales de las variables a predecir con los valores de las predicciones, de manera que cuanto más se parezcan mejor será el modelo.

- El *aprendizaje no supervisado* es un tipo de aprendizaje automático en que no existen un conjunto de variables a predecir y la máquina se dedica a aprender patrones que tengan los datos. Por tanto, como esos patrones de los datos no se saben "a priori", ya que no son variables de los datos, no se puede evaluar la calidad de los modelos comparándolos con unos valores "reales", sino que la evaluación de la calidad de los modelos se centra en medir cómo de bien los patrones obtenidos explican los datos.

Los métodos de aprendizaje automático supervisado se pueden dividir a su vez según el tipo de variable dependiente:

- *Regresión*: La variable a predecir es cuantitativa, i.e., numérica (tanto continua como discreta).
- *Clasificación*: La variable a predecir es cualitativa y por tanto no se pueden hacer operaciones matemáticas con ella. En algún caso las variables cualitativas se sustituyen por el valor de un número (etiquetas numéricas), pero al representar categorías (raza, religión, estado civil, ...) no se pueden hacer operaciones con ellas. En caso de que la variable a predecir sea dicotómica (tenga solo dos valores o categorías posibles) estamos ante clasificación dicotómica o binaria.

En este trabajo emplearemos métodos supervisados de clasificación dicotómica. En la segunda parte del método descrito en el capítulo 3 se emplea una máquina vector soporte, que explicaremos en la sección 2.4. En la segunda parte del método descrito en el capítulo 4, que describiremos en la sección 2.5. En cada método se ha usado uno de esos métodos de aprendizaje automático porque eran los mejores (según el procedimiento de la sección 2.3.2) de un conjunto de métodos candidatos.

Antes de ello vamos a explicar cuestiones que son comunes a los métodos de clasificación dicotómicos.

### 2.3.1 Sobreajuste, división de conjuntos, validación cruzada.

Cuando trabajamos con un conjunto de datos, ese conjunto de datos suele ser una muestra de una población (por ejemplo: los resultados de una encuesta es una muestra respecto de la población total a la que se dirigía la encuesta) y por tanto la información que saquemos de los datos vamos a tomarla como una hipótesis de toda la población. Por lo tanto, queremos que los modelos que obtengamos de los datos también funcionen en la población. Aquí es donde entra el sobreajuste, que significa que el modelo funciona muy bien en el conjunto de datos con el que la máquina aprende, pero no tan bien en el total de la población.

Para evitar que el modelo que seleccionemos (en este caso de clasificación dicotómica) se ajuste muy bien a los datos pero no se pueda generalizar, dividimos el conjunto de datos que trabajamos en dos de manera aleatoria:

- *Conjunto de entrenamiento*: Suele contener entre el 70 y el 80 por ciento de los datos. Sobre este conjunto se construyen los modelos.
- *Conjunto test*: Suele contener entre el 20 y el 30 por ciento de los datos. Sobre este conjunto se evalúan los modelos construidos sobre el conjunto de entrenamiento.

Así, con esta división no solo buscamos un modelo que se ajuste bien a los datos con los que se construye, sino que se pueda generalizar. La capacidad de generalización la medimos en el conjunto test, ya que ese conjunto no se ha usado para construir el modelo.

Cuando construimos un modelo a partir de un método de aprendizaje automático, ese modelo no solo va a depender de los datos del conjunto de entrenamiento sino que puede depender de uno o más hiperparámetros, como por ejemplo en una regresión polinomial sería el grado del polinomio del modelo o en una regresión múltiple cuantas variables tiene el modelo. Para escoger estos hiperparámetros tenemos un problema parecido al que planteábamos antes, saber a partir del conjunto de entrenamiento cuáles serán los mejores hiperparámetros en el conjunto test. Además muchas veces con unos valores para los hiperparámetros obtenemos un modelo muy complejo que se sobreajusta al conjunto de entrenamiento (se ajusta muy bien al conjunto de entrenamiento pero tiene peor rendimiento en el conjunto test) y con otros valores obtenemos modelos muy simples que no tienen buen rendimiento ni en el conjunto de entrenamiento ni en el test.

Un método que se emplea para escoger los hiperparámetros correctos es la *validación cruzada*. Con este método se divide el conjunto de entrenamiento  $C$  en  $k$  partes  $C_1, \dots, C_k$  de igual tamaño. Nos quedamos con unos posibles valores para los hiperparámetros a escoger. Para cada parte  $i$  del conjunto de entrenamiento,  $i \in \{1, \dots, k\}$  construimos sobre  $C \setminus C_i$  un modelo para cada valor posible de los hiperparámetros y evaluamos el modelo en el conjunto de datos  $C_i$ . Finalmente, nos quedaremos con los valores para los hiperparámetros que mejor evaluación obtenga (considerando la media de las evaluaciones en los conjuntos  $C_1, \dots, C_k$ ). Normalmente, se suelen considerar  $k = 5$  o  $10$ .

Cómo evaluar un modelo de clasificación binaria se verá en la subsección 2.3.3.

### 2.3.2 Elección del mejor método de aprendizaje automático con validación cruzada anidada LTO

Supongamos que tenemos varios métodos de aprendizaje automático, y queremos escoger que método funciona mejor con un tipo de datos que tenemos. Para ello, vamos a realizar un tipo de validación cruzada algo distinta al de la sección 2.3.1 en el que vamos a testear los métodos sobre todo el conjunto. El método que vamos a usar, propuesto por Thi Lan Anh Dinh et al. en [17], es la validación cruzada anidada dejando dos fuera (*leave-two-out*, LTO).

Primero, dividimos el conjunto de datos en  $k$  subconjuntos del mismo tamaño:  $C_1, \dots, C_k$ . Después, para cada iteración  $i \in \{1, \dots, k-1\}$ , construimos un modelo en el conjunto de entrenamiento  $C_1 \cup \dots \cup C_{i-1} \cup C_{i+1} \cup \dots \cup C_k$  con los hiperparámetros que mejor evaluación obtenga en el conjunto de validación  $C_{i+1}$  y evaluamos ese modelo en el conjunto test  $C_i$ . En la iteración  $i = k$  hacemos lo mismo solo que con el conjunto de entrenamiento  $C_2 \cup \dots \cup C_{k-1}$ , de validación  $C_k$  y test  $C_1$ . Nos quedamos con el método que obtenga la mejor evaluación media. Observar que cada modelo que construimos y evaluamos en cada iteración se construye usando hiperparámetros fijados únicamente para esa iteración. Se suele considerar  $k = 5$  o  $10$ .

Una vez hayamos escogido el mejor método para nuestros datos, podemos considerar, para dicho método, la validación cruzada (VC) explicada en la subsección 2.3.1, i.e., dividimos en conjunto test y entrenamiento una única vez y fijamos los hiperparámetros solo una vez por VC dividiendo el conjunto de entrenamiento. Así, obtendríamos el método entrenado con unos hiperparámetros fijos.

Cuadro 2.1: Una matriz de confusión para la clasificación  $\hat{y}$  de  $y$ .

		Valores $y = 1$	Reales $y = -1$	
Valores	$\hat{y} = 1$	Verdaderos Pos. (VP)	Falsos Pos. (FP)	VP+FP
Predichos	$\hat{y} = -1$	Falsos Neg. (FN)	Verdaderos Neg. (VN)	FN+VN
		VP+FN	FP+VN	n=VP+FP+FN+VN

### 2.3.3 Evaluación de la calidad de una clasificación supervisada binaria

Sea un conjunto de datos  $(x_1, y_1), \dots, (x_n, y_n)$ , con  $x_i \in \mathbb{R}^p$  e  $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$  donde las primeras  $p$  coordenadas son las variables predictoras y la última coordenada es la variable dicotómica  $y$  a predecir. Supongamos que hemos aplicado un método de aprendizaje automático en el que hemos obtenido una predicción  $\hat{y}$  de los valores de la variable  $y$ , denotando para cada  $i \in \{1, \dots, n\}$  la predicción de  $y_i$  como  $\hat{y}_i$ , entonces un primer método de evaluar la calidad de la clasificación  $\hat{y}$  de  $y$  es la realización de una tabla de confusión.

La *matriz de confusión* consiste en una matriz que cuenta el número de veces que  $\hat{y}$  toma cada valor de  $\{-1, +1\}$  cuando  $y$  vale un valor  $\{-1, +1\}$ . Es decir, una matriz con la frecuencia absoluta conjunta de  $y$  e  $\hat{y}$ . En el cuadro 2.1 podemos ver como son las matrices de confusión y lo que significa cada elemento de la matriz. Así, una clasificación binaria será mejor cuanto los elementos que tiene en la diagonal sean más grandes y más pequeños los elementos fuera de la diagonal.

**Definición 2.19.** *A partir de la tabla de confusión se pueden calcular la siguientes medidas de la bondad de la clasificación:*

- *Exactitud (o accuracy en inglés): Proporción de datos clasificados correctamente, i.e., la suma de la diagonal de la matriz de confusión entre la suma del total de elementos de la matriz:*

$$AC := \frac{VP + VN}{n}.$$

- *Sensibilidad: Proporción de positivos (en este caso de datos con  $y = 1$ ) clasificados correctamente, i.e.,*

$$SE := \frac{VP}{VP + FN}.$$

Cuadro 2.2: Matriz de confusión del ejemplo 2.4.

		Valores reales		
		$y = 1$	$y = -1$	
Valores predichos	$\hat{y} = 1$	90	10	100
	$\hat{y} = -1$	0	0	0
		90	10	100

- *Precisión: Proporción de datos positivos entre el total de los clasificados como positivos, i.e.,*

$$PR := \frac{VP}{VP + FP}.$$

- *Especificidad: Proporción de negativos (en este caso de datos con  $y = -1$ ) clasificados correctamente, i.e.,*

$$ES := \frac{VN}{FP + VN}.$$

Todas estas medidas valen siempre entre 0 y 1 y el modelo será mejor cuando estas medidas más se acerquen a 1. La medida o medidas que son más apropiadas para valorar el modelo depende del objetivo que tengamos con el modelo (clasificar bien los datos en general, clasificar bien los positivos/negativos, tener pocos falsos positivos, ...) Puede ser que dos medidas nos den un resultado muy bueno pero, sin embargo, otra medida tenga un resultado muy pobre. Esto puede ocurrir cuando el número de elementos de una clase y otro son muy distintos. Lo podemos ver en este simple ejemplo.

*Ejemplo 2.4.* Un método de clasificación que tenga la matriz de confusión del cuadro 2.2 tiene:

$$AC = 0.9, SE = 1 \text{ y } ES = 0.$$

Que se clasifiquen tan mal los negativos no afecta mucho a la proporción total de datos clasificados correctamente (exactitud), ya que solo el 10 por ciento de los datos son negativos.

## 2.4 Máquina Vector Soporte

Una *máquina vector soporte* (MVS) es un método de aprendizaje automático que clasifica una variable de un conjunto de datos  $(p + 1)$ -dimensionales según si el vector  $p$ -dimensional de sus variables predictoras estén a un lado u otro de un hiperplano (óptimo). Esto significa que, dado un conjunto de datos  $(x_1, y_1), \dots, (x_n, y_n)$ , con  $x_i \in \mathbb{R}^p$  e  $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$ , la MVS construye una forma de predecir si  $y_i = +1$  o  $-1$ , en función de los valores de su vector  $x_i$  y, para ello, separaremos los vectores  $x_i$  en dos grupos separados por un hiperplano en  $\mathbb{R}^p$ .

La máquina vector soporte se define inicialmente para clasificar una variable dicotómica, pero se puede generalizar a variables cuantitativas. En este trabajo consideraremos solo variables dicotómicas donde  $y_i = 1$  si  $i$  pertenece a la clase  $C_1$  o  $y_i = -1$  si  $i$  pertenece a la clase  $C_2$ . Primero, tratamos el caso de que los datos sean separables linealmente en  $\mathbb{R}^p$  según su clase, luego, generalizaremos a datos casi linealmente separables introduciendo términos de error, y finalmente, generalizaremos el modelo para poder hacer separaciones no lineales proyectando los datos a dimensiones superiores.

### 2.4.1 Datos separables linealmente

Sean  $(x_1, y_1), \dots, (x_n, y_n)$ , con  $x_i \in \mathbb{R}^p$  e  $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$ . Si suponemos que los datos se pueden separar linealmente, entonces existe al menos un hiperplano  $x^t \beta + \beta_0 = 0$  con  $\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}$  tal que para todo  $i \in \{1, \dots, n\}$  se cumple que o bien  $x_i^t \beta + \beta_0 \geq 0$  si  $y = 1$  o bien  $x_i^t \beta + \beta_0 \leq 0$  e  $y_i = -1$ . Ahora, vamos a intentar que los datos estén lo más separados del hiperplano separador. Dado que la distancia de un dato  $x_i$  al hiperplano  $x^t \beta + \beta_0 = 0$  es:

$$\frac{|x_i^t \beta + \beta_0|}{\|\beta\|};$$

si  $y = 1$ , como  $x_i^t \beta + \beta_0 \geq 0$ , vamos a intentar maximizar  $\frac{x_i^t \beta + \beta_0}{\|\beta\|}$  y si  $y = -1$ , como  $x_i^t \beta + \beta_0 \leq 0$ , vamos a intentar minimizar  $\frac{x_i^t \beta + \beta_0}{\|\beta\|}$ . Entonces obtenemos el siguiente problema de optimización:



$$\begin{aligned} & \underset{\|\beta\|=1}{\text{máx}} M \\ \text{s.a. } & y_i(x_i^t\beta + \beta_0) \geq M \quad \forall i \in \{1, \dots, n\} \end{aligned} \quad (2.1)$$

Con este problema estamos intentando maximizar la distancia del dato más cercano al hiperplano. Como los vectores normalizados en  $\mathbb{R}^p$  incluyen todas las direcciones que hay en  $\mathbb{R}^p$  podemos restringirnos al caso  $\|\beta\| = 1$  sin pérdida de generalidad.

Si consideramos todos los vectores  $\beta \in \mathbb{R}^p$  sin normalizar modificando los  $\beta_0$  por  $\|\beta\|\beta_0$  las restricciones del problema son equivalentes a

$$\frac{y_i}{\|\beta\|}(x_i^t\beta + \beta_0) \geq M \quad \forall i \in \{1, \dots, n\}.$$

Estas condiciones son equivalentes a  $y_i(x_i^t\beta + \beta_0) \geq M\|\beta\|$ . Como dados unos  $\beta \in \mathbb{R}^p$ ,  $\beta_0 \in \mathbb{R}$  que satisfagan la última desigualdad  $\forall i \in \{1, \dots, n\}$ , si los multiplicamos por un número real positivo cualquiera se siguen satisfaciendo las desigualdades y por tanto podemos tomar arbitrariamente  $\|\beta\| = \frac{1}{M}$ . Así, reescalando otra vez el  $\beta_0$ , aplicando sobre la función objetivo la función monótona decreciente  $f(x) = \frac{1}{2x^2}$  y cambiando maximizar por minimizar, (2.1) es equivalente a:

$$\begin{aligned} & \underset{\|\beta\|=1}{\text{mín}} \frac{1}{2}\|\beta\|^2 \\ \text{s.a. } & y_i(x_i^t\beta + \beta_0) \geq 1 \quad \forall i \in \{1, \dots, n\} \end{aligned} \quad (2.2)$$

Con las restricciones estamos creando un par de rectas paralelas  $x^t\beta + \beta_0 = 1$  y  $x^t\beta + \beta_0 = -1$  donde si  $y_i = 1$  entonces  $x_i^t\beta + \beta_0 \geq 1$  y  $y_i = -1$  entonces  $x_i^t\beta + \beta_0 \leq -1$  para todo  $i$ . Es decir, estamos creando dos rectas paralelas donde no haya ningún dato y donde al otro lado de cada recta solo haya puntos de una clase. La distancia entre esas rectas paralelas, el ancho de ese margen entre clases, es  $\frac{2}{\|\beta\|}$ , que es lo que estamos maximizando al minimizar  $\|\beta\|^2$ .

## 2.4.2 Datos cuasi-separables linealmente

En el caso de que los conjuntos no sean separables linealmente, vamos a introducir al problema (2.2) un término de error para cada punto  $i \in \{1, \dots, n\}$ :

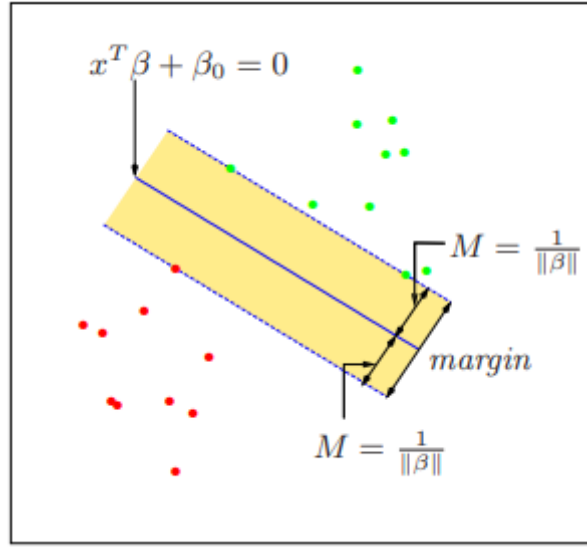


Figura 2.8: Cuando los datos son separables linealmente obtenemos que podemos separar los datos de las variables regresoras según a la clase a la que pertenezcan dividiendo el espacio por un hiperplano. Además, no solo podemos separarlas por un hiperplano  $x^t \beta + \beta_0 = 0$ , si no que podemos hacer que los puntos de una clase cumplan  $x^t \beta + \beta_0 > 1$  y que los de la otra  $x^t \beta + \beta_0 < -1$ , separándolas por un margen de ancho  $\frac{2}{\|\beta\|}$ . En la imagen podemos ver esto último, donde los puntos correspondientes a una clase son de color rojo y los de la otra son de color verde. Las rectas discontinuas son  $x^t \beta + \beta_0 = 1$  y  $x^t \beta + \beta_0 = -1$ . Imagen sacada de [19].

$$\epsilon_i = \begin{cases} 0 & \text{si } y_i(x_i^t \beta + \beta_0) \geq 1 \\ |y_i - (x_i^t \beta + \beta_0)| & \text{en caso contrario.} \end{cases}$$

Así, dado un  $i \in \{1, \dots, n\}$  su variable de holgura puede tomar los siguientes valores:

- $\epsilon_i > 1$  si el dato está clasificado mal, i.e., si  $y_i(x_i^t \beta + \beta_0) < 0$ ;
- $0 < \epsilon_i \leq 1$  si está clasificado bien pero dentro de hiperplanos márgenes, i.e., si  $0 \leq y_i(x_i^t \beta + \beta_0) < 1$ ; y
- $\epsilon_i = 0$  si cumple las condiciones de (2.2), i.e., si  $y_i(x_i^t \beta + \beta_0) \geq 1$ .

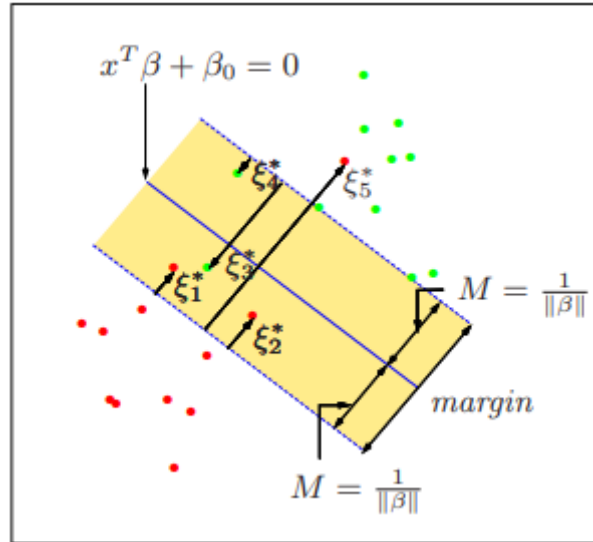


Figura 2.9: En la imagen vemos una generalización de la gráfica de la figura 2.8 introduciendo los términos del error. Así  $0 < \xi_1^*, \xi_2^*, \xi_4^* < 1$  ya que los puntos se encuentran bien clasificados (en el lado correcto respecto de la recta  $x^T \beta + \beta_0 = 0$ );  $\xi_3^*, \xi_5^* > 1$ , ya que se encuentran mal clasificados, de hecho  $\xi_3^* < 2 < \xi_5^*$ , puesto que el primero está mal clasificado pero dentro del margen y el segundo no está dentro del margen. Imagen sacada de [19].

Por tanto el problema a resolver es el siguiente (con  $C > 0$ ):

$$\begin{aligned} & \min \left( \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \epsilon_i \right) \\ \text{s.a. } & y_i (x_i^t \beta + \beta_0) \geq 1 - \epsilon_i \quad \forall i \in \{1, \dots, n\} \\ & \epsilon_i \geq 0 \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

Estamos por una parte maximizando el margen (al minimizar  $\|\beta\|^2$ ) pero minimizando los errores. El parámetro  $C > 0$  mide cuanto penalizamos el error respecto de lo que "premiarnos" el margen. Así, cuando más grande sea  $C > 0$  obtendremos errores  $\epsilon_i$  más pequeños pero márgenes de separación también más pequeños, y cuando más pequeño sea  $C$  obtendremos márgenes de separación mayores pero también errores más grandes. Por tanto este parámetro es un hiperparámetro que se ha de fijar con validación cruzada para ni ajustarnos demasiado ni ajustarnos poco al conjunto de entrenamiento.

La función (primal) de Lagrange que se obtiene es:

$$L_p = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \epsilon_i - \sum_{i=1}^n \alpha_i [y_i(x_i^t \beta + \beta_0) - (1 - \epsilon_i)] - \sum_{i=1}^n \mu_i \epsilon_i,$$

siendo los  $\alpha_i \geq 0$  los multiplicadores de Lagrange correspondientes a las primeras  $n$  restricciones y los  $\mu_i \geq 0$  los correspondientes a las segundas  $n$  restricciones. Por tanto, tenemos que minimizar esta función con respecto a  $\beta$ ,  $\beta_0$  y  $\epsilon := (\epsilon_1, \dots, \epsilon_n)$ . Si derivamos la función respecto de estas variables e igualamos a 0 obtenemos

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i \quad (2.3)$$

$$0 = \sum_{i=1}^n \alpha_i y_i \quad (2.4)$$

$$\alpha_i = C - \mu_i \quad \forall i \in \{1, \dots, n\} \quad (2.5)$$

además de las restricciones de positividad:  $\alpha_i, \mu_i, \epsilon_i \geq 0 \quad \forall i \in \{1, \dots, n\}$ .

Sustituyendo (2.3), (2.4) y (2.5) en la función primal de Lagrange, obtenemos la función objetivo dual de Lagrange:

$$L_d = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^t x_j.$$

Esta función nos da una cota inferior a la función objetivo primal,  $L_p$ . El problema dual entonces es encontrar la mejor cota inferior, nos queda:

$$\begin{aligned} & \text{máx} \quad L_d \\ & \text{s.a.} \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\} \\ & \quad \sum_{i=1}^n \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned} \quad (2.6)$$

Además de las ecuaciones (2.3), (2.4) y (2.5), las condiciones de optimalidad de Karush-Kuhn-Tucker las restricciones:

$$\alpha_i [y_i(x_i^t \beta + \beta_0) - (1 - \epsilon_i)] = 0,$$

$$\begin{aligned}\mu_i \epsilon_i &= 0, \\ y_i(x_i^t \beta + \beta_0) - (1 - \epsilon_i) &\geq 0,\end{aligned}$$

$\forall i \in \{1, \dots, n\}$ . Estas desigualdades junto con las ya mencionadas (2.3), (2.4) y (2.5) caracterizan de manera única la solución al problema primal y dual.

Una vez obtenidas las soluciones  $\hat{\beta}_0$  y  $\hat{\beta}$  clasificamos un punto  $x_i \in \mathbb{R}^p$  como  $y_i = 1$  si  $x_i^t \hat{\beta} + \hat{\beta}_0 > 0$  y como  $y_i = -1$  si  $x_i^t \hat{\beta} + \hat{\beta}_0 < 0$ .

### 2.4.3 Datos no linealmente separables

En caso de que los datos  $(x_1, y_1), \dots, (x_n, y_n)$ , con  $x_i \in \mathbb{R}^p$  e  $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$  no sean linealmente separables (ni cuasi-linealmente) según el valor de la variable  $y$ , lo que podemos hacer es proyectar las variables explicativas a un espacio de dimensión mayor donde los datos sí que sean linealmente separables. Es decir, encontrar una función  $\phi: \mathbb{R}^p \rightarrow \mathbb{R}^q$  con  $q \geq p$  tal que los datos  $(\phi(x_1), y_1), \dots, (\phi(x_n), y_n)$ , con  $\phi(x_i) \in \mathbb{R}^q$  e  $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$  sean linealmente separables (o cuasi-linealmente) y resolver el problema de optimización presentado en el apartado anterior.

Actuando como en el problema anterior obtenemos el siguiente problema dual:

$$\begin{aligned}\text{máx} \quad L_d &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(x_i)^t \phi(x_j) \\ \text{s.a.} \quad 0 &\leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n \sum_{i=1}^n \alpha_i y_i &= 0\end{aligned}\tag{2.7}$$

y una vez resuelto obtenemos clasificamos según el valor de  $\phi(x)^t \beta + \beta_0 = \sum_{i=1}^n \alpha_i y_i \phi(x)^t \phi(x_i)$  (hemos usado 2.3). Tanto en la resolución del problema dual como en la regla de clasificación no se emplea el valor de las proyecciones  $\phi(x)$  sino el producto escalar entre ellas  $\phi(x_i)^t \phi(x_j)$ . Por lo tanto, para reducir costes computacionales, en la práctica se emplean funciones núcleo

$$K: (x, y) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto K(x, y) = \phi(x)^t \phi(y) \in \mathbb{R}$$

directamente, y así se evita proyectar a espacios superiores para luego calcular el producto escalar.

## 2.5 KNN, método de los K Vecinos más Cercanos

El método de los  $K$  Vecinos más Cercanos (*K Nearest Neighbours*, KNN) es un método de aprendizaje automático sencillo que clasifica una variable cualitativa  $Y$  de un conjunto de datos  $(p + 1)$ -dimensionales empleando una distancia en el espacio dónde están las variables predictoras (llamadas  $X$ ),  $\mathbb{R}^p$ . Fijada una distancia, normalmente la euclidiana, este método depende únicamente de un hiperparámetro,  $K \in \mathbb{N}$ .

Dada una observación del conjunto test  $x_0 \in \mathbb{R}^p$  (formado por las  $p$  coordenadas que se usan para predecir), el KNN identifica los  $K$  datos del conjunto de entrenamiento más cercanos (en el espacio de las variables predictoras)  $\mathcal{N}_0$ , el conjunto de los  $K$  vecinos más cercanos. Después, estima  $\hat{Y}_0$ , la clase a la que corresponde el dato (i.e. el valor de la variable  $Y$ ), de la siguiente forma:

$$\hat{Y}_0 = \arg \max_{j \in T} \sum_{i \in \mathcal{N}_0} I(y_i = j),$$

donde  $T$  es el conjunto formado por las clases posibles, i.e., es el conjunto formado por los posibles valores de la variable cualitativa a predecir  $Y$ ;  $y_i$  es el valor de la variable  $Y$  en la observación  $i \in \mathcal{N}_0$  y la función  $I$  es una función indicatriz que nos indica si el dato  $i \in \mathcal{N}_0$  pertenece a la clase  $j \in T$ , i.e.:

$$I(y_i = j) = \begin{cases} 1 & \text{si } y_i = j \\ 0 & \text{si } y_i \neq j \end{cases}$$

Resumidamente, se predice la clase a la que pertenece un dato del conjunto test como la clase mayoritaria en los  $k$  datos del conjunto de entrenamiento más cercanos (en el espacio dónde están las variables predictoras).

El valor de  $K$  escogido tiene un efecto drástico en el modelo clasificador obtenido. Si  $K$  es muy pequeño, el clasificador es muy flexible (produciendo “fronteras de decisión”<sup>2</sup> muy ajustadas a cada dato) y se ajusta demasiado a los datos de entrenamiento, encontrando patrones en el conjunto de entrenamiento que no

---

<sup>2</sup>Una frontera de decisión es la frontera entre los conjuntos formados por los puntos que si fuesen del conjunto test se clasificarían a una clase concreta.

existen. Por lo tanto, estamos ante un clasificador con poco sesgo pero mucha varianza, puesto que un ligero cambio en los datos pueden cambiar bastante el modelo. Si  $K$  es demasiado grande, el método es demasiado poco flexible y produce una “frontera de decisión” casi lineal. Estaríamos entonces ante un clasificador con mucho sesgo y poca varianza, ya que los cambios en los datos afectan poco al modelo. En la figura 2.10 podemos ver un ejemplo de esto en una simulación de una clasificación binaria, probando el mismo modelo con  $K = 1$  y con  $K = 100$ ; resultando un modelo muy ajustado a los datos y un modelo poco ajustado a los datos, respectivamente.

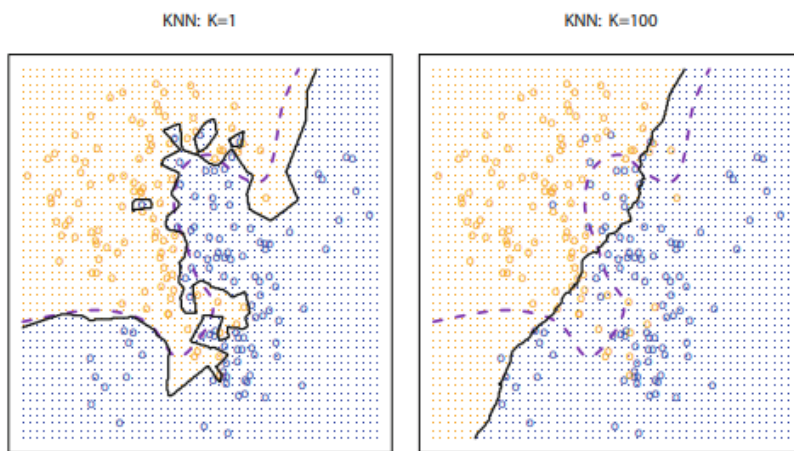


Figura 2.10: Comparación de las fronteras de decisión (representadas por líneas continuas) de un modelo KNN binario entrenado con datos simulados con 100 observaciones (los puntos pertenecen a la clase según su color). El color de la rejilla del fondo indica la clase que el modelo asignaría a ese punto de la rejilla si estuviese en un conjunto test. En la izquierda se ha usado  $K = 1$ , obteniendo una frontera de decisión demasiado flexible y sobreajustándose a los datos de entrenamiento y a la derecha se corresponde con  $K = 100$  teniéndose el efecto contrario. Imagen sacada de [18].





## 3 | Estado del arte

En este capítulo describiremos el método de Muszynski, G. et al. ([7]) para detectar si existe o no un río atmosférico en una región de la Tierra en un determinado momento de tiempo. Este método extrae patrones topológicos sobre los datos, en particular, la evolución de las componentes conexas del subgrafo de nivel superior. Luego, con aprendizaje automático, se intenta diferenciar qué patrones se corresponden con ser RA, dando resultado a una clasificación de las imágenes como RA o como no RA. Para el aprendizaje automático se emplean unas etiquetas provenientes de otro método, que también se usan para medir la calidad de la clasificación.

En la sección 3.1 se describen con qué conjuntos de datos se ha trabajado y en la sección 3.2 se muestra un esquema del método. A continuación, detallamos la primera etapa del método, donde se usa ATD, en la sección 3.3 y detallamos la segunda etapa, la del aprendizaje automático, en la sección 3.4. Finalmente, se muestran y analizan los resultados que se obtienen en la sección 3.5.

### 3.1 Datos empleados

Los conjuntos de datos que se emplean vienen tanto de la simulación de modelos climáticos generada por la versión 5.1 de la Community Atmosphere Model (CAM5.1) ([20]), como del producto del reanálisis de la segunda versión del Modern-Era Retrospective Analysis for Research and Applications (MERRA-2) ([21]). El reanálisis es un proceso a través del cual un método fijo de asimilación de datos (que depende de un modelo meteorológico de predicción) se usa para adquirir reprocesamientos consistentes de observaciones meteorológicas. De estos datos se emplea la columna que nos indica el DSVa.

Cuadro 3.1: Tabla con los datos que se emplean, tanto provenientes de simulaciones como de reanálisis, y con sus períodos y resoluciones.

Modelo climático	Período	Resolución temporal	Resolución espacial
CAM5.1	1979 – 2005	3h y 24h	25 km
CAM5.1	1979 – 2005	3h y 24h	100 km
CAM5.1	1979 – 2005	3h y 24h	200 km
MERRA-2	1980 – 2017	3h	50 km

Aunque TDLVA es otra variable también usada en la identificación de ríos atmosféricos, no es observable por satélite, al contrario que la DSVa. Sin embargo, aunque no se contempla en esta memoria, también se pueden hacer modelos de identificación de RA basados en TDLVA.

Estos modelos dan imágenes en escala de grises de dimensión 2, que son observaciones en un instante de tiempo del valor de DSVa a lo largo de una rejilla que tiene puntos a lo largo de un mapa de una región. Estas imágenes en escala de grises vienen representadas por matrices. En este caso se emplea como región el área del pacífico que va desde las islas de Hawái en EE. UU. hasta la costa oeste de los EE. UU. (la región que se ve en la figura 1.1). Se utilizan datos con distintas precisiones temporales (distancia de tiempo entre los instantes de tiempo en los que observamos la variable) y espaciales (distancia entre los puntos de la rejilla contiguos). Los modelos de datos empleados, precisiones y períodos de tiempo correspondientes se pueden ver en el cuadro 3.1.

Para que se vea el formato y el tamaño de los datos, veamos una submatriz  $7 \times 7$  de una matriz ( $73 \times 112$ ) de uno de los conjuntos usados, el proveniente del MERRA-2 con precisión espacial de  $50km$  y temporal de  $3h$ :

$$\begin{pmatrix} 23.658932 & 23.036139 & 22.123043 & 21.430202 & 22.051882 & 23.914698 & 25.141624 \\ 21.240788 & 20.877882 & 21.092453 & 22.048023 & 23.707134 & 25.040762 & 25.717901 \\ 20.433863 & 21.380583 & 22.717028 & 24.17173 & 25.391476 & 25.826097 & 25.939281 \\ 22.098997 & 23.790562 & 25.248415 & 25.969824 & 26.210562 & 26.03678 & 25.719465 \\ 24.092237 & 25.286072 & 26.315365 & 26.481539 & 26.273636 & 26.1369 & 26.369896 \\ 24.930016 & 25.768875 & 26.599392 & 26.673508 & 26.62879 & 27.229403 & 28.029497 \\ 25.454836 & 25.905527 & 26.493378 & 27.133888 & 28.030231 & 28.798422 & 28.613619 \end{pmatrix}$$

Por otra parte, como se tiene que entrenar un clasificador de aprendizaje automático no supervisado, se necesita que las observaciones estén etiquetadas como RA o no RA, es decir, sobre si en cada imagen en escala de grises hay un río atmosférico o no. Las etiquetas se obtienen aplicando a los datos el modelo *Toolkit for Extreme Climate events Analysis* (TECA) ([22]). Este método clasifica según un criterio fijo basado en un umbral. Se asume que las etiquetas que aporta la TECA como "etiquetas verdaderas".

Los datos que se usan están disponibles en [23].

## 3.2 Esquema del método

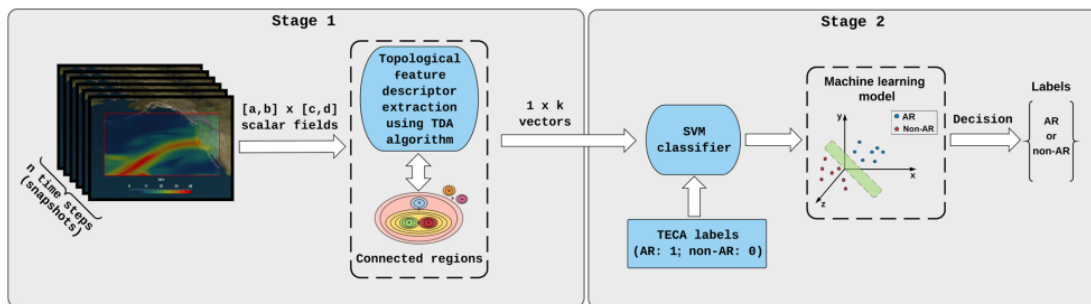


Figura 3.1: Esquema del método. Imagen sacada de [7].

El método consta de dos etapas:

- Etapa 1: Se extraen características topológicas de las imágenes con Análisis Topológico de Datos. Se emplea el algoritmo Union-Find para sacar información sobre la evolución de las componentes conexas en unos conjuntos de nivel. Estas características topológicas serán el *input* del clasificador de la segunda etapa.
- Etapa 2: Clasificamos las características previas con un método aprendizaje automático, la Máquina Vector Soporte (MVS). La clasificación se divide en dos pasos: Primero, se entrena el modelo para separar las características topológicas en dos clases (RA y no RA) en el conjunto de entrenamiento y después se le aplica el modelo al conjunto test. Posteriormente, se evalúa

la clasificación en términos de exactitud, sensibilidad y precisión. Como estamos realizando aprendizaje supervisado, tanto para entrenar el modelo como para evaluar sus resultados, vamos a emplear las etiquetas que nos aporta el modelo TECA.

El método tiene los siguientes datos de entrada y de salida.

- Datos de entrada:  $n$  imágenes en escalas de grises bidimensionales con datos en  $n$  instantes de tiempo sobre la densidad superficial de vapor de agua (DSVA) en  $kgm^{-2}$ . Cada imagen en escala de grises viene representada por una matriz. Para cada imagen tenemos una etiqueta (las obtenidas del modelo TECA) que nos indica a que clase pertenece (RA o no RA) esa observación.
- Datos de salida. Etiquetas binarias. Si es RA,  $RA = 1$ , si no lo es,  $RA = 0$ .

### 3.3 Primera etapa, análisis topológico de datos

En esta etapa solo se emplean las imágenes sin las etiquetas (es decir, las coordenadas y el valor de DSVA). Así, los datos se pueden representar como una función definida en una rejilla, determinada por las coordenadas de los datos, al conjunto de los reales, donde la imagen de las coordenadas es el valor del DSVA en ese punto. Como el valor de DSVA es positivo y nunca supera su máximo  $L$  en la rejilla (existe el máximo en la rejilla al ser un conjunto finito de puntos) entonces esa función tiene su imagen dentro de  $[0, L]$ , donde en este caso  $L = 60kgm^{-2}$

Esta función sería la restricción de una función

$$DSVA : [a, b] \times [c, d] \longrightarrow [0, L']$$

definida en toda la región del estudio (y no únicamente en la rejilla) donde  $a$ ,  $b$ ,  $c$  y  $d$  representarían los límites izquierdo/derecho/superior/inferior de la región y donde la función asigna a cada punto su valor de la densidad superficial de vapor de agua.  $L'$  sería el valor máximo de DSVA en la región.

Por otra parte, podemos modelar la rejilla como un grafo donde cada punto está unido por aristas a los puntos que están inmediatamente a su izquierda,

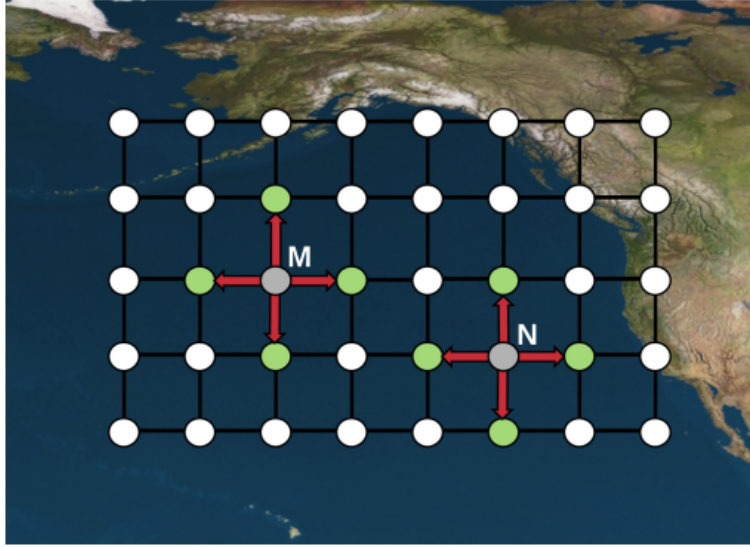


Figura 3.2: Representación de la rejilla como un grafo donde cada punto tendría como aristas los vecinos inmediatamente superiores, inferiores, a su izquierda y derecha. Imagen sacada de [7].

derecha, arriba y abajo. Así, cada punto tendría 4 vecinos (salvo los puntos de la frontera). Podemos ver una representación del grafo en la figura 3.2.

Definimos el conjunto de nivel superior de  $f$  (en la región  $[a, b] \times [c, d]$ ) como los puntos de la región que tienen DSVA mayor que un valor  $t$ :

$$(\text{DSVA})^{-1}[t, +\infty) = \{(x, y) \in [a, b] \times [c, d] | f(x, y) \geq t\}.$$

Interesa saber como evoluciona la forma de las zonas de la región con DSVA mayor que cierto umbral a medida que va disminuyendo el umbral, ya que los ríos atmosféricos tienen una forma característica (son bandas alargadas y estrechas dónde hay un alto valor de DSVA). En particular, se va a ver como evolucionan las componentes conexas de  $(\text{DSVA})^{-1}[t, +\infty)$ .

Como solo se saben los valores en la rejilla, se va a ver como evoluciona la conexidad de los subconjuntos de nivel superior de DSVA restringida a la rejilla. Es decir, son subgrafos dónde se considera que:

- Un nodo está en el subconjunto de nivel si su valor de DSVA es mayor o igual que el umbral.

- Una arista está en un subconjunto de nivel superior si lo están los dos puntos (nodos) de la rejilla que une.

En el grafo se considera que dos puntos están en la misma componente conexa si existe un camino que los une.

Se va a calcular el conjunto de nivel superior de  $f$  en la rejilla para los valores  $t \in \{1, 1 + p, 1 + 2p, \dots, L = 60 \text{ kgm}^{-2}\}$  para el umbral (siendo  $p > 0$  un paso entre umbrales considerados) y se va a medir cómo evolucionan las componentes conexas del conjunto según vayamos reduciendo el valor del umbral. Cuando  $t > L$  este conjunto es vacío, cuando  $t$  se va reduciendo aparecen componentes conexas que a medida que se sigue reduciendo  $t$  se van a sumergir unas en otras.

Solo interesan las componentes conexas que intersequen ciertas subregiones del mapa. En el caso particular de este estudio, como se está tratando en el pacífico noreste, solamente interesan las componentes conexas que intersequen la latitud de Hawái y el continente americano. En la figura 3.3 se pueden ver las subregiones que interesan (la costa americana rodeada de verde y las latitudes de Hawái en amarillo). Si alguna componente conexa no interseca la subregión de interés, se actúa como si no existiese esa componente conexa, ya que esa componente conexa no puede formar un patrón de río atmosférico, ya que no va desde los trópicos hasta el continente.

### 3.3.1 Gráficos de evolución

Para obtener las características topológicas que luego se usan como *input* en la máquina vector soporte se emplea un algoritmo basado en el algoritmo *Union-Find data structure* ([24]).

Con este algoritmo se obtiene el número de puntos de la rejilla que están en una componente conexa a medida que el umbral disminuya (se toman como valores para el umbral  $1, 1 + p, 1 + 2p, \dots, L = 60 \text{ kgm}^{-2}, p > 0$ )<sup>1</sup>.

Para determinar en qué componente conexa se cuentan los puntos de la rejilla se emplea la **Elder Rule**. La *Elder Rule* consiste en que cuando dos componentes conexas se unen, se considera que la que ha aparecido antes (la que apareció

---

<sup>1</sup>En la implementación hecha por Muszynski, G. et al. se ha considerado un paso  $p = 1$ .

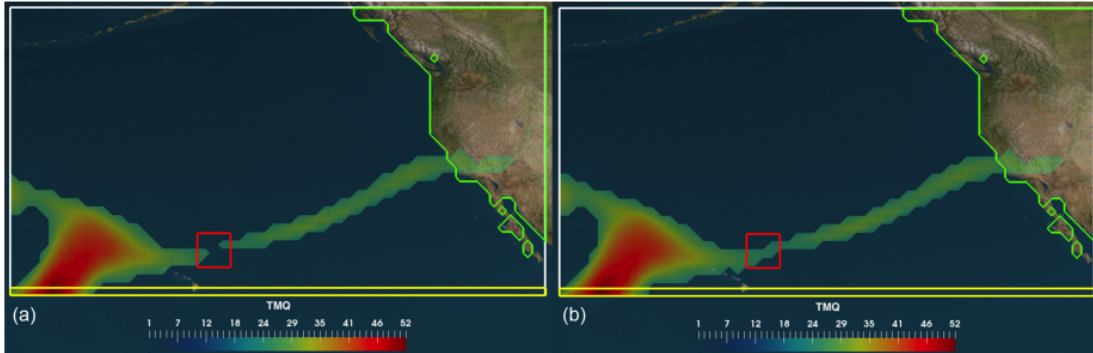


Figura 3.3: Una ilustración de la búsqueda de regiones conectadas de RA en unas subregiones específicas de interés. En este ejemplo, las subregiones que interesan son la costa americana rodeada de verde y las latitudes de Hawái en amarillo. (a) El cuadrado rojo indica el lugar donde dos componentes conexas del conjunto de nivel superior no están conectadas para un valor del umbral ( $DSVA = t_1$ ). (b) Con nuevo umbral ( $DSVA = t_2$ ), donde  $t_2 < t_1$ , las dos componentes conexas se unen haciendo que el conjunto de nivel superior tenga forma de RA. El DSV A (en  $kgm^{-2}$ ) mostrado en este ejemplo proviene del modelo CAM5.1. Imagen sacada de [7].

con un umbral más grande) es la que absorbe a la otra y sobrevive. Por tanto, con la *Elder Rule* estamos midiendo el número de elementos que tiene para cada umbral la primera componente que aparece (ya que es la única que cuando dos componentes conexas se unen siempre sobrevive). El algoritmo descarta las componentes conexas que no interseca las subregiones de interés y no las emplea en sus cálculos.

Esta información que nos aporta el algoritmo la podemos representar en gráficos de evolución. Los gráficos muestran el número de puntos rejilla en una componente conexa (la que aparece antes) según el valor de DSV A. Estos gráficos muestran el número normalizado de puntos de la rejilla, es decir, el porcentaje de puntos de la rejilla, que están en una componente conexa para cada valor umbral. Se normalizan los puntos para así poder comparar gráficos de rejillas con distinta resolución espacial (por tanto, distinto número total de puntos en la rejilla).

En la figura 3.4 podemos ver tres gráficos de evolución para el mismo conjunto de datos: a la izquierda podemos ver los gráficos de evolución de 100 observaciones etiquetadas como ríos atmosféricos y la media de todas las observaciones

etiquetadas como RA; en el centro los gráficos de evolución de 100 observaciones etiquetadas como no RA y la media de todas las observaciones etiquetadas como no RA; y en el centro se comparan las dos medias previas. Estos gráficos medios son muy parecidos, casi indiferenciables para un ojo humano, pero son distintos, y es algo de lo que se encargará después la máquina vector soporte.

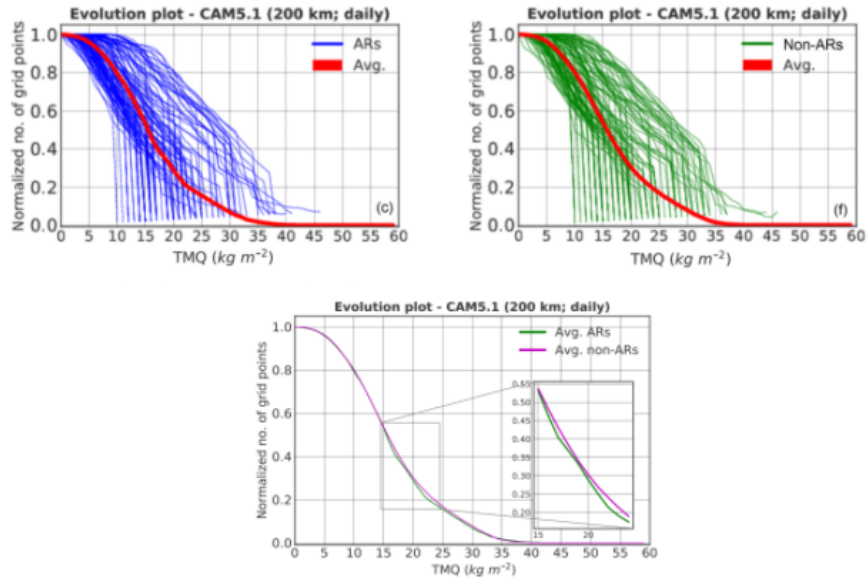


Figura 3.4: Gráficos de evolución normalizados de 100 observaciones y la media (rojo) de todas las observaciones. En la parte de arriba, a la izquierda, vemos el grafo de evolución para observaciones etiquetadas por TECA como RA (azul) y, en la derecha, para etiquetadas como no RA. En la parte de abajo, comparamos ambas medias. Las observaciones se corresponden con los datos sacados del modelo de simulación CAM 5.1 con precisión espacial de  $200km$  y observaciones diarias. Imagen sacada de [7].

De estos gráficos de evolución se puede extraer por cada observación un vector  $k$  dimensional donde la coordenada  $i$ -ésima del vector se corresponde con el número normalizado de puntos en una componente conexas considerando el umbral  $i$ -ésimo. Así, se obtendría, juntando todas las observaciones, una matriz con  $n$  filas y  $k$  columnas, donde  $n$  es el número de observaciones (instantes de tiempo en el que se tiene una imagen de la variable DSVA en el mapa) y  $k$  es el número de umbrales que se consideran (por tanto, el tamaño de las características



topológicas extraídas de cada observación). Esta matriz es el *input* de la máquina vector soporte descrito en la siguiente sección. En este caso, como tenemos que  $L = 60 \text{kgm}^{-2}$  es el valor máximo de DSVA en la rejilla, se consideran los umbrales  $1, 2, \dots, 60 = L$ , y por tanto, se considera  $k = 60$  y un paso entre umbrales de  $p = 1$ . Si redujésemos el paso, aumentaríamos la dimensión de los datos. En la figura 3.5 podemos ver un esquema de cómo se genera la matriz *input*.

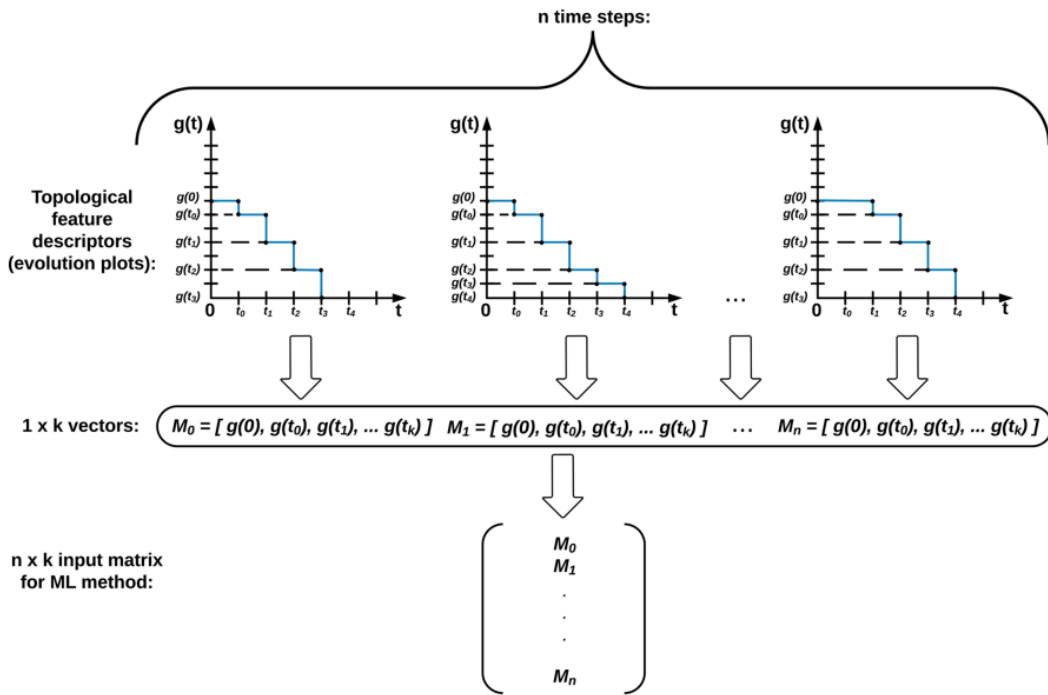


Figura 3.5: Esquema de cómo se crea la matriz *input* para la MVS. Imagen sacada de [7].

### 3.3.2 Ejemplo simple del funcionamiento del algoritmo

En esta sección vamos a mostrar un ejemplo simple de como funciona el algoritmo. Vamos a suponer que nuestra matriz de datos (que nos da el valor de DSVA a lo largo de la rejilla) es una matriz  $5 \times 5$ . Modelamos entonces la rejilla como un grafo con 25 nodos que están unidos por aristas con los nodos justamente inferiores, superiores, a su izquierda y a su derecha. Además, por simplificar el

ejemplo vamos a rastrear las componentes conexas de los subconjuntos de nivel superior (i.e., subgrafos formados por nodos donde el valor de DSVA es mayor que un umbral dado y por las aristas del grafo que unen nodos que están el subgrafo) solo para los umbrales 15, 20, 25, 30, 35, 40, 45. Como estamos considerando 7 umbrales el resultado que vamos a obtener es un vector con 7 coordenadas, donde cada una nos indica la proporción de nodos que están en la componente conexa más vieja de subconjuntos de nivel superior, empezando por 45.

En la figura 3.6 podemos ver: la matriz de valores de DSVA a lo largo de la rejilla *input*, el grafo modelando la rejilla con los datos de la matriz (donde los colores indican las subregiones de interés) y el *output* del algoritmo (e *input* de la máquina vector soporte de la siguiente sección), i.e., la proporción de nodos que están en la componente conexa más "vieja" de subconjuntos de nivel superior (empezando por 45).

En las figuras 3.7 y 3.8 vemos los subconjuntos de nivel superior para los umbrales considerados. A partir de estos subconjuntos se obtiene el vector de la figura 3.6 siguiendo estos pasos:

- Hasta el umbral 35, incluido, no se considera que exista ninguna componente conexa (y por ello las 3 primeras coordenadas del vector resultado son 0). Esto es debido a que, si bien en los umbrales 40 y 35 existe una componente conexa, esta no se cuenta, ya que no toca las dos subregiones de interés (la zona violeta y la zona naranja).
- En el umbral 30, la componente conexa que no tocaba antes (con umbrales superiores) las subregiones de interés, ahora toca ambas, formando un patrón de RA por lo que ya se considera como componente conexa. Como es la primera componente conexa que aparece en los subgrafos, vamos, a partir del umbral 30, a contar cuantos elementos (en proporción con el total) tiene esa componente a medida que vayamos a reducir el umbral y por ello las últimas cuatro coordenadas del vector resultado son  $(\frac{5}{25}, \frac{9}{25}, \frac{12}{25}, \frac{25}{25})$ . Destacar, que a pesar de que para los umbrales 25 y 20 aparezca una componente conexa distinta, ni la componente ni el punto que forma parte de ella se tienen en cuenta, ya que solo se rastrea a la componente conexa que aparezca antes tocando ambas subregiones de interés.

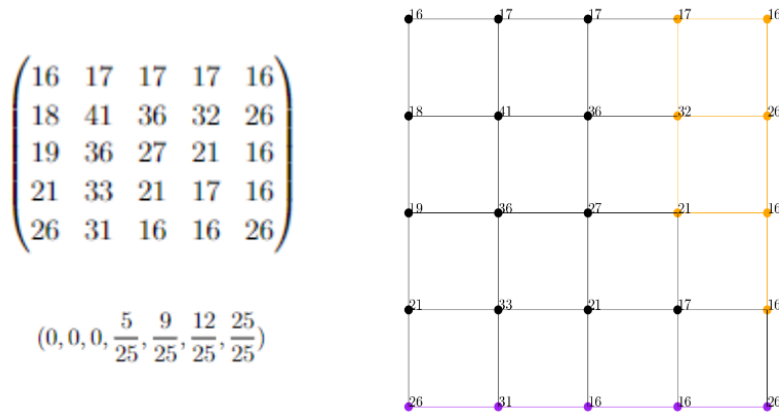


Figura 3.6: Arriba a la izquierda podemos una matriz de valores ejemplo de DSVA a lo largo de la rejilla. A la derecha un grafo modelando esa rejilla (con los valores de DSVA asociados a los nodos en pequeño) donde las subregiones de interés están coloreadas (la zona naranja representa América y la zona violeta representa las latitudes de Hawái). Abajo a la izquierda vemos el resultado de aplicar el algoritmo.

### 3.4 Segunda etapa, clasificación con máquina vector soporte

Después de la extracción de características topológicas, para cada imagen se tiene, por un lado, un vector  $k$  dimensional con el número de elementos de una componente conexa para los distintos valores del umbral, y por el otro lado, las clasificaciones de las imágenes obtenidas al aplicarles el modelo TECA.

Con esos datos se pone en marcha un método de clasificación supervisada donde las  $k$  variables predictoras son las correspondientes a las características topológicas extraídas previamente y la variable a predecir (a clasificar, en este caso, al ser variable cualitativa) es la variable  $RA$  (valdría 1 si en la observación hay un río atmosférico y  $-1$  en caso contrario). Para más información y detalle sobre los métodos de clasificación supervisada, ver la sección 2.3.

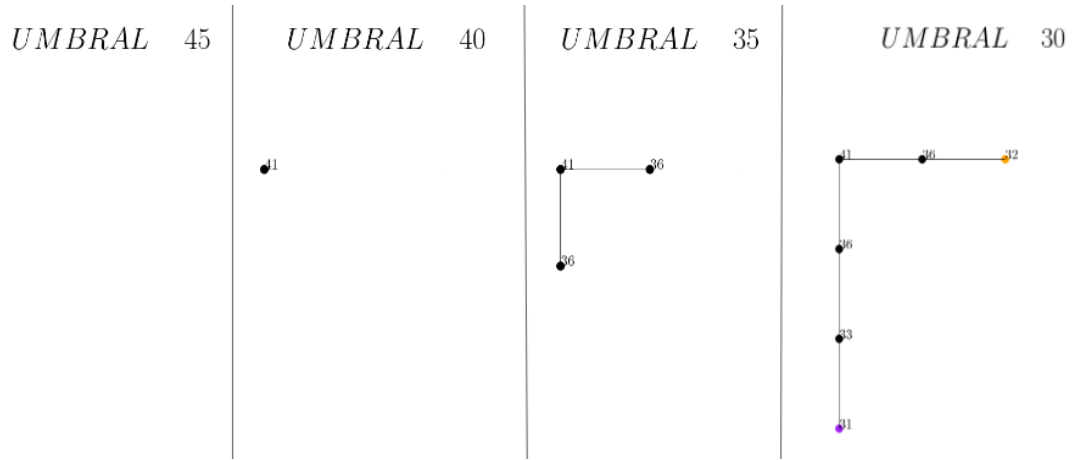


Figura 3.7: Conjuntos de nivel superior para distintos umbrales para el ejemplo de la subsección 3.3.2, representado además en la figura 3.6.

Siguiendo el procedimiento descrito en la subsección 2.3.2, se ha determinado que el mejor modelo (entre un conjunto de modelos de aprendizaje automático candidatos) para clasificar es una Máquina Vector Soporte (MVS) con función núcleo de base radial:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|), \quad \gamma > 0.$$

Con este método se proyectan los vectores  $k$  dimensionales de las variables predictoras (la información topológica) a un espacio de dimensión superior a través de una función  $\phi: \mathbb{R}^k \rightarrow \mathbb{R}^p$ ,  $p > r$  y se clasifican una observación como RA o no RA según a qué lado de un hiperplano óptimo esté la imagen por la proyección del vector con la información topológica. El hiperplano óptimo está determinado por la resolución de este problema de optimización para los  $m$  datos del conjunto de entrenamiento:

$$\begin{aligned} & \min\left(\frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^n \epsilon_i\right) \\ \text{s.a. } & y_i(\phi(x_i)^t \beta + \beta_0) \geq 1 - \epsilon_i \quad \forall i \in \{1, \dots, m\} \\ & \epsilon_i \geq 0 \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

donde  $x_1, \dots, x_m \in \mathbb{R}^k$  son los vectores de las variables predictoras para cada observación del conjunto de entrenamiento. Para resolver este problema se usa que

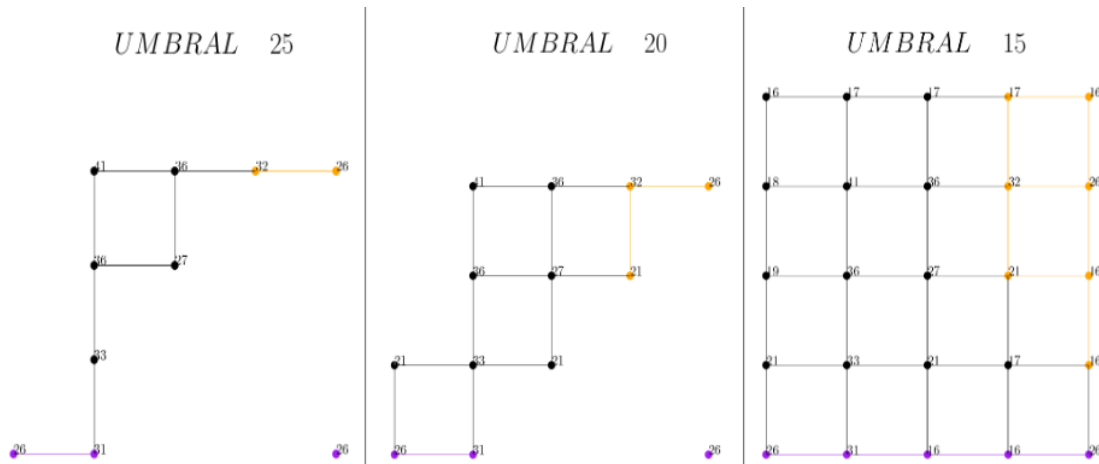


Figura 3.8: Conjuntos de nivel superior para distintos umbrales para el ejemplo de la sección 3.3.2, representado además en la figura 3.6.

$K(x, y) = \phi(x)^t \phi(y) \forall x, y \in \mathbb{R}^k$  y así no se tienen que calcular las proyecciones  $\phi(x_i)$  (ya que la solución al problema solo depende de los productos escalares de las proyecciones). Para más información y detalle sobre la máquina vector soporte y los núcleos, consultar la sección 2.4.

Para determinar el valor óptimo para  $C$  y  $\gamma$  se emplea validación cruzada optimizando la exactitud (detalles sobre validación cruzada para un método fijo en la subsección 2.3.1) en dos pasos. Primero se prueba con valores para  $(C, \gamma)$  en una rejilla “con mucho hueco”, es decir, con bastante diferencia entre dos puntos de la rejilla consecutivos. Después se afina la rejilla (se reduce la diferencia entre dos valores de la rejilla consecutivos) alrededor de los puntos donde la exactitud es más alta y se concreta más los valores de los dos parámetros.

*Observación 3.1.* Al igual que otros métodos de aprendizaje automático, la MVS, tiende a sobreajustarse a la clase mayoritaria, por tanto, tiende a clasificar bastante bien los datos de la clase mayoritaria y mal los datos de la clase minoritaria. Para resolver este problema se ha aplicado remuestreo para obtener conjuntos de entrenamiento con clases equilibradas. El remuestreo se ha realizado antes de la división en conjunto de entrenamiento y test.

Este remuestreo también se ha hecho en la validación cruzada anidada LTO previa en la que se ha seleccionado el mejor método de aprendizaje automático.

## 3.5 Resultados

La parte de análisis topológica de datos da unos resultados que se pueden representar en gráficos de evolución (ver sección 3.3.1) como el que se puede ver en la figura 3.4. En esta figura podemos ver cómo los gráficos de evolución medios de datos etiquetados como RA son casi indistinguibles de los gráficos de evolución medios de datos etiquetados como no RA. Veremos ahora que la MVS es capaz de detectar estas pequeñas diferencias para clasificar bastante bien los gráficos de evolución en RA o no RA.

Una vez aplicada la MVS a los gráficos de evolución, se mide la calidad de su clasificación en tres de las medidas de bondad que se han definido en la sección 2.3.3: exactitud, precisión y sensibilidad. Estas tres medidas se miden (valga la redundancia) en el conjunto test.

Podemos ver en el cuadro 3.2 la exactitud obtenida en el conjunto test para los distintos conjuntos de datos. Si nos fijamos en la exactitud respecto de la precisión espacial, vemos cómo se tiene más exactitud cuanto menos resolución haya. En [7] se especula con que cuanto más resolución en los conjuntos de datos tiende a haber más ruido, obteniendo representaciones topológicas menos suaves y empeorando la exactitud. Si nos fijamos en la exactitud respecto de la precisión temporal, vemos como reducir la precisión temporal y, por tanto, reducir el número de imágenes con los que entrenar una MVS, no empeora la exactitud de la clasificación. Esto sugiere que la representación topológica en precisión diaria contiene suficiente información sobre lo que es RA y no RA para que la MVS sea capaz de distinguir entre las dos clases con una exactitud alta. Finalmente, vemos que la exactitud con los datos de MERRA es similar a la exactitud con los datos con precisión horaria de  $3h$ , indicando que el método de detección de RA es robusto con respecto a la fuente de datos de DSVa.

Podemos ver en el cuadro 3.3 la precisión y sensibilidad obtenida en el conjunto test para los distintos conjuntos de datos. Podemos apreciar como la precisión es mayor a la sensibilidad para todos los conjuntos de datos. Esto indica que el porcentaje de error en las imágenes clasificadas como RA es menor que el porcentaje de error en las imágenes que son RA.

Cuadro 3.2: Tabla con la exactitud en el conjunto test de la MVS, para los distintos conjuntos de datos considerados.

Mod. climático	Resol. temporal	Resol. espacial	Exactitud
CAM5.1	3h	25 km	0.83
CAM5.1	3h	100 km	0.77
CAM5.1	3h	200 km	0.90
CAM5.1	24h	25 km	0.82
CAM5.1	24h	100 km	0.84
CAM5.1	24h	200 km	0.91
MERRA-2	3h	50 km	0.80

Cuadro 3.3: Tabla con la precisión y sensibilidad en el conjunto test de la MVS, para los distintos conjuntos de datos considerados.

Mod. climático	Resol. temporal	Resol. espacial	Precisión	Sensibilidad
CAM5.1	3h	25 km	0.91	0.74
CAM5.1	3h	100 km	0.83	0.67
CAM5.1	3h	200 km	0.95	0.85
CAM5.1	24h	25 km	0.87	0.77
CAM5.1	24h	100 km	0.86	0.83
CAM5.1	24h	200 km	0.97	0.85
MERRA-2	3h	50 km	0.84	0.74

En resumen, el modelo de detección de RA obtiene una alta exactitud (0.77-0.91) aplicándolo a diversos conjuntos de datos con distinto origen, precisión temporal y precisión espacial. Esto ilustra que la combinación de análisis topológico de datos y aprendizaje automático es una estrategia efectiva y eficiente para detectar RA en conjuntos de datos climáticos grandes sin tener que fijar previamente umbrales subjetivos. Además, el modelo de aprendizaje automático está sesgado por las etiquetas empleadas provenientes de TECA, ya que hemos asumido que estas etiquetas son "verdad" y hemos entrenado y calificado nuestro modelo con ellas. Queda pendiente de comprobar y usar el método con distintos etiquetados provenientes de otros métodos.



## 4 | Método propuesto

En este capítulo proponemos un método que mejora el de Muszynski, G. et al. [7]. En el capítulo 5 compararemos los resultados y veremos que el método que proponemos mejora en términos de estabilidad y de resultados numéricos.

Al igual que el método de Muszynski, G. et al. con este método vemos si existe o no un río atmosférico en una región de la Tierra en un determinado momento de tiempo. Este método extrae patrones topológicos sobre los datos, en particular, la evolución de las componentes conexas y de los huecos en los complejos cúbicos filtrados asociados a las imágenes en escala de grises. Luego, con aprendizaje automático, se intenta diferenciar qué patrones se corresponden con ser RA, dando resultado a una clasificación de las imágenes como RA o como no RA. Para el aprendizaje automático se usan unas etiquetas provenientes de otro método, que también se usan para medir la calidad de la clasificación.

La principal diferencia de nuestro método con el de Muszynski, G. et al. es que no solo extraemos información sobre la evolución de las componentes conexas sino que también sobre la evolución de los huecos. Es por ello que esperamos que mejore los resultados de Muszynski, G. et al.. Además: las etiquetas que usamos en el aprendizaje automático (provenientes de TECA-BARD) son distintas a las que usan Muszynski, G. et al. en su método (provenientes de TECA) y empleamos otro método de aprendizaje automático.

En la sección 4.1 se describen con qué conjuntos de datos se ha trabajado y en la sección 4.2 mostramos un esquema del método propuesto y qué diferencias y similitudes tiene con el anterior. A continuación, detallamos la primera etapa de nuestro método, donde se usa ATD, en la sección 4.3 y detallamos la segunda etapa, la del aprendizaje automático, en la sección 4.4. Finalmente, hablamos sobre el código y la implementación a ordenador en 4.5 y los resultados que se obtienen se muestran y analizan en 4.6.

Cuadro 4.1: Tabla con los datos que se emplean, provenientes de simulaciones como de reanálisis, y con sus períodos y resoluciones.

Modelo climático	Período	Resolución temporal	Resolución espacial
MERRA-2	1980 – 2017	3h y 24 h	50 km

## 4.1 Datos empleados

Los conjuntos de datos que se emplean vienen del producto del reanálisis de la segunda versión del Modern-Era Retrospective Analysis for Research and Applications (MERRA-2) ([21]). El reanálisis es un proceso a través del cual un método fijo de asimilación de datos (que depende de un modelo meteorológico de predicción) se usa para adquirir reprocesamientos consistentes de observaciones meteorológicas.

Al igual que en el capítulo anterior, de estos datos se emplea la columna que nos indica el DSVA y no se utiliza TDLVA, que es otra variable que se emplea en la detección de ríos atmosféricos, ya que no es observable por satélite, al contrario que la DSVA.

Este modelo da imágenes en escala de grises de dimensión 2, que son observaciones en un instante de tiempo del valor de DSVA a lo largo de una rejilla que tiene puntos a lo largo de un mapa de una región. Estas imágenes en escala de grises vienen representadas por matrices. Al igual que en el método descrito en el capítulo 4 se usa como región el área del pacífico que va desde las islas de Hawái en EE. UU. hasta la costa oeste de los EE. UU. (la región que se ve en la figura 1.1). Se emplean datos con distintas precisiones temporales (distancia de tiempo entre los instantes de tiempo en los que observamos la variable). Los modelos de datos empleados, precisiones y periodos de tiempo correspondientes se pueden ver en el cuadro 4.1.

Para que se vea el formato y el tamaño de los datos provenientes del MERRA-2, veamos una submatriz  $7 \times 7$  de una matriz ( $73 \times 112$ ) de los conjuntos utilizados:

$$\begin{pmatrix} 23.658932 & 23.036139 & 22.123043 & 21.430202 & 22.051882 & 23.914698 & 25.141624 \\ 21.240788 & 20.877882 & 21.092453 & 22.048023 & 23.707134 & 25.040762 & 25.717901 \\ 20.433863 & 21.380583 & 22.717028 & 24.17173 & 25.391476 & 25.826097 & 25.939281 \\ 22.098997 & 23.790562 & 25.248415 & 25.969824 & 26.210562 & 26.03678 & 25.719465 \\ 24.092237 & 25.286072 & 26.315365 & 26.481539 & 26.273636 & 26.1369 & 26.369896 \\ 24.930016 & 25.768875 & 26.599392 & 26.673508 & 26.62879 & 27.229403 & 28.029497 \\ 25.454836 & 25.905527 & 26.493378 & 27.133888 & 28.030231 & 28.798422 & 28.613619 \end{pmatrix}$$

Por otra parte, como se tiene que entrenar un clasificador de aprendizaje automático no supervisado, se necesita que las observaciones estén etiquetadas como RA o no RA, es decir, sobre si en cada imagen hay un río atmosférico o no. Las etiquetas se obtienen aplicando a los datos el modelo *TECA Bayesian Atmospheric River Detector* (TECA-BARD) ([25]). Este método automático de detección de RA se ha construido a partir de datos etiquetados manualmente por expertos. Se asume que las etiquetas que aporta el TECA-BARD como "etiquetas verdaderas".

En comparación, vamos a usar un conjunto de datos de los que usan Muszynski, G. et al. en su método ([7]) y además vamos a usar ese conjunto de datos, pero con solo un dato cada día. Sin embargo, no se van a emplear las mismas etiquetas para ese conjunto de datos, sino que las etiquetas que vamos a utilizar son provenientes de un método distinto. Esta diferencia simplemente es debida a las dificultades de acceso a los datos. A pesar de ello, los resultados son comparables pero teniendo en cuenta ese matiz.

Los datos que se utilizan están disponibles en [23]. Las etiquetas para los datos empleados <sup>1</sup> están disponibles en el siguiente link: [https://www.earthsystemgrid.org/dataset/ucar.cgd.cesm4.artmip.tier1.cascade\\_bard\\_v1.atm.other.3hourly\\_inst.html](https://www.earthsystemgrid.org/dataset/ucar.cgd.cesm4.artmip.tier1.cascade_bard_v1.atm.other.3hourly_inst.html)

---

<sup>1</sup>Se han procesado los datos de una rejilla a lo largo de la región (por cada instante de tiempo) con 1 en los puntos dónde había un río atmosférico y 0 si no. Hemos obtenido, a partir de cada matriz (rejilla) un único número por cada rejilla que nos indicase si en la región de estudio había un río atmosférico que tocase las latitudes de Hawái y el continente americano.

## 4.2 Esquema del método y diferencias con el método existente

El esquema general de nuestro método es parecido al de Muszynski, G. et al., primero realizamos análisis topológico de datos (ATD) y extraemos un vector por cada imagen con información topológica y luego clasificamos esos vectores con aprendizaje automático. Además, estamos realizando el método para el Pacífico noreste, pero este método es adaptable a otras regiones (al igual que con el método anterior). La diferencia reside en que empleamos otro método de ATD y de aprendizaje automático.

El método consta de dos etapas:

- Etapa 1: Se extraen características topológicas de las imágenes con Análisis Topológico de Datos. Se emplea homología persistente en complejos cúbicos filtrados con el fin de extraer información sobre la evolución de las componentes conexas y los huecos. Estas características topológicas serán el *input* del clasificador de la segunda etapa.
- Etapa 2: Clasificamos las características previas con un método aprendizaje automático, los K Vecinos más Cercanos (*K Nearest Neighbours*, KNN). La clasificación se divide en dos pasos: Primero se entrena el modelo para separar las características topológicas en dos clases (RA y no RA) en el conjunto de entrenamiento y después se le aplica el modelo al conjunto test. Posteriormente, se evalúa la clasificación en términos de exactitud, sensibilidad y precisión. Como estamos realizando aprendizaje supervisado, tanto para entrenar el modelo como para evaluar sus resultados, vamos a emplear las etiquetas que nos aporta el modelo TECA-BARD.

El método tiene los siguientes datos de entrada y de salida.

- Datos de entrada:  $n$  imágenes en escalas de grises bidimensionales con datos en  $n$  instantes de tiempo sobre la densidad superficial de vapor de agua (DSVA) en  $kgm^{-2}$ . Cada imagen en escala de grises viene representada por una matriz. Para cada imagen tenemos una etiqueta (las obtenidas del modelo TECA-BARD) que nos indica a que clase pertenece (RA o no RA) esa observación.

- Datos de salida. Etiquetas binarias. Si es  $RA$ ,  $RA = 1$ , si no lo es,  $RA = 0$ .

Así las tres diferencias de nuestro método con el de Muszynski, G. et al. son:

1. Se emplea homología persistente en dimensiones 0 y 1 en vez de emplear el algoritmo Union-Find para rastrear la evolución de las componentes conexas en los subconjuntos de nivel superior. Con homología persistente en dimensiones 0 y 1 se rastrea la evolución de las componentes conexas y de los huecos en los subconjuntos de nivel superior. El objetivo de este cambio es detectar, además de la evolución de las componentes conexas, posibles patrones de huecos en los subconjuntos de nivel superior de los RA que los diferencien.  
Además, con el algoritmo Union-Find se saca distinto tipo de información sobre las componentes conexas (tamaño de la componente conexas más vieja) que la información que se saca con homología persistente (información relacionada con el nacimiento y muerte de todas las componentes)
2. Se emplea KNN en vez de MVS como método de aprendizaje automático. Esto es debido a que KNN da resultados ligeramente mejores.
3. Se emplean etiquetas surgidas del modelo TECA-BARD y no del TECA. Esta diferencia simplemente es debida a las dificultades de acceso a los datos. Queda pendiente (al igual que en el método de Muszynski, G. et al.) probar el método con etiquetas provenientes de otras fuentes.

### 4.3 Primera etapa, análisis topológico de datos

En esta etapa solo se emplean las imágenes sin las etiquetas (es decir, las coordenadas y el valor de DSVA). Vamos a intentar extraer información sobre la estructura topológica de los datos que sea relevante para que, en la siguiente etapa, el método de aprendizaje automático sepa diferenciar entre RA y no RA.

Como hemos dicho anteriormente, vamos a usar homología persistente en complejos cúbicos filtrados para extraer información sobre la evolución de componentes conexas y de huecos en los datos. Una manera directa de hacer esto es modelar las imágenes en escala de grises con el valor de DSVA como complejos cúbicos filtrados (hemos explicado en la subsección 2.1.3 como se hace) y aplicar homología persistente en esos complejos cúbicos (explicada en la subsección 2.1.4).

Sin embargo, no vamos a aplicar esto directamente sobre las imágenes con el valor de DSVA sino que vamos a modificar esas imágenes. Vamos a explicar primero lo que se va a aplicar sobre esas imágenes modificadas y luego podremos explicar (en la subsección 4.3.1) porque se ha aplicado sobre esas imágenes modificadas y no sobre las originales.

Supongamos que tenemos esas imágenes ya modificadas respecto de las originales con el valor de DSVA. Como las imágenes modificadas no van a ser muy distintas, podemos pensar como que estas imágenes modificadas nos indican igualmente la concentración de humedad a lo largo de la región.

Nos interesa saber como evoluciona la forma de las zonas de la región con una concentración de humedad mayor que cierto umbral a medida que va disminuyendo este umbral. Esto es debido a que los ríos atmosféricos, que son zonas de la superficie de la tierra con alta concentración de humedad, tienen una forma característica: son bandas alargadas y estrechas. Así, estamos rastreando qué forma tienen estas regiones (con concentración de humedad mayor que cierto umbral) y para qué umbral se obtiene cada forma (ya que no es lo mismo una banda fina de humedad con un umbral medio-bajo que con un umbral alto). La manera de medir como evoluciona la forma de estas zonas de la región es rastreando la evolución de las componentes conexas y huecos (de estas zonas).

La manera de medir esta evolución es modelando estas imágenes modificadas como complejos cúbicos filtrados y aplicar homología persistente en ellos. La forma de modelar un complejo cúbico filtrado está explicada en la sección 2.1.3 y la homología persistente en complejos cúbicos en la sección 2.1.4. Con la homología persistente vamos a rastrear el nacimiento y muerte de clases de equivalencia en los grupos de homología de dimensión 1 y 0 según vayamos moviendo el umbral en los subcomplejos de nivel superior. A partir de la homología persistente vamos a obtener dos diagramas de persistencia:

- El diagrama de persistencia de dimensión 0, con la información sobre el nacimiento y muerte de clases de equivalencia en los grupos de homología de dimensión 0. Equivalentemente, el diagrama de persistencia de dimensión 0 nos da información sobre nacimiento y muerte de **componentes conexas** en los subcomplejos de nivel superior.
- El diagrama de persistencia de dimensión 1, con la información sobre el nacimiento y muerte de clases de equivalencia en los grupos de homología de

dimensión 1. Equivalentemente, el diagrama de persistencia de dimensión 0 nos da información sobre nacimiento y muerte de **huecos** en los subcomplejos de nivel superior.

Estos diagramas de persistencia se han calculado en una implementación en ordenador, de manera que no va a haber puntos del diagrama con multiplicidad mayor que 1 salvo que haya varias clases que nazcan en el mismo umbral.

*Observación 4.1.* En la subsección 2.1.4 se han definido las filtraciones (def. 2.13) empleando subcomplejos de nivel inferior, mientras que nosotros queremos ver como evolucionan los subcomplejos de nivel superior (definiendo las filtraciones de esa manera). Existen dos soluciones posibles:

- La primera es que, para adaptarse a las definiciones que hemos hecho en el capítulo de conceptos previos, considerar, en vez de las matrices que representan las imágenes modificadas, las matrices multiplicadas por  $-1$  al modelar los complejos cúbicos filtrados. Así, al calcular los subcomplejos de nivel **inferior** respecto la función de filtrado asociada a la matriz multiplicada por  $-1$  estamos calculando los subcomplejos de nivel **superior** de respecto la función de filtrado asociada a la matriz original, ya que es claro que:

$$f^{-1}([r, +\infty)) = (-f)^{-1}((-\infty, r]).$$

Esta es la solución que se emplea en la implementación a ordenador

- La segunda manera sería considerar las filtraciones en la definición 2.13 como subconjuntos de nivel superior. Además, para que estos subconjuntos de nivel superior sean subcomplejos deberíamos cambiar en la definición de función monótona (def. 2.9)  $f(P) \leq f(Q)$  por  $f(P) \geq f(Q)$  y en la construcción T (def. 2.12) el mínimo por el máximo al definir la función de filtrado en cubos de dimensión inferior a la máxima.

Esta es la solución que se aplica cuando se hacen a mano ejemplos, como en el ejemplo de la subsección 4.3.2 (salvo el diagrama de persistencia).

### 4.3.1 Obtención de función de filtrado

Veamos ahora por qué tenemos que aplicar homología persistente sobre esas matrices modificadas y no sobre las originales.

El motivo es porque la homología persistente no tiene en cuenta que los ríos atmosféricos tienen que ir desde las latitudes de Hawái hasta el continente americano. Estamos tratando de igual forma a una componente que va desde/hasta alguna de esas regiones que a una completamente alejada de ambas. Para cambiar esto, necesitamos no tener en cuenta en la información topológica extraída componentes conexas irrelevantes para la detección de RAs, como también se hizo en el método de Muszynski, G. et al.

Para no tener en cuenta componentes conexas irrelevantes para la detección de RAs, se modifican las imágenes en escala de grises de manera que al aplicar homología persistente no entren componentes conexas que no toquen ninguna de las dos subregiones de interés (latitudes de Hawái o el continente americano). Como la conexión en grafos es equivalente a la conexión en complejos cúbicos y como existen muchos algoritmos implementados en ordenador para rastrear los elementos de las componentes conexas de un grafo, vamos a calcular las imágenes en escala de grises modificadas modelando la rejilla donde tenemos datos como un grafo de igual manera que se hizo en el método de Muszynski, G. et al.. Es decir, cada punto de la imagen (o elemento de la matriz) es un nodo del grafo que está unido por aristas a los puntos que están inmediatamente a su izquierda, derecha, arriba y abajo. Podemos ver una representación de grafo subyacente a la rejilla en la figura 3.2.

En el algoritmo que presentamos en esta subsección, vamos a cambiar la matriz con los valores originales de DSVA por unos valores *modificados* donde no va a haber para ningún umbral componentes conexas en el subcomplejo cúbico de nivel superior (estricto) que no toquen los puntos de la rejilla que se corresponden con el continente americano o las latitudes de Hawái. Para ello, los valores *modificados* van a ser el primer umbral para el cual ese punto de la rejilla entra en el conjunto de nivel superior (estricto) en una componente conexa que toque algún punto correspondiente con América o las latitudes de Hawái.

Concretamente, el algoritmo que aplicamos a cada imagen en escala de grises (es decir, una matriz) que tenemos en los datos <sup>2</sup> es el siguiente.

Sea  $G$  el grafo subyacente a la rejilla. Sea  $A$  la matriz con los valores de DSVA en los puntos de la rejilla

$$N = \left\{ (i, j) \mid i \in \{1, \dots, n\}, j \in \{1, \dots, m\} \right\},$$

---

<sup>2</sup>Dada una matriz  $A \in M_{n,m}(\mathbb{R})$ , denotamos el elemento en la posición  $(i, j)$  como  $A_{ij}$ .



siendo  $N$  los nodos de  $G$ . Sea  $AH \subset N$  los puntos de la rejilla que se corresponden con el continente Americano o con las latitudes de Hawái.  $F$  es la matriz que obtenemos como resultado, la imagen *modificada*. Sea  $p > 0$  un paso.<sup>3</sup>

- El *input* al algoritmo es  $A$ .
- Inicializamos la matriz  $F = A$  función de filtrado resultado y la matriz: indicador acumulado  $Indacum = 0_{n,m} \in M_{n,m}(\mathbb{R})$ .
- Para cada umbral  $u \in U := \{60, 60 - 1p, 60 - 2p, \dots, 0\}$ <sup>4</sup> en orden descendente:
  1. Inicializamos la matriz indicador  $Indi = 0_{n,m} \in M_{n,m}(\mathbb{R})$ .
  2. Calculamos las componentes conexas  $Comp$  del subgrafo  $Subgrf$  que tiene los nodos  $N_{Subgrf} = \{(i, j) \in N | A_{ij} > u\}$  y las aristas que tiene son las de  $G$  que unen nodos de  $N_{Subgrf}$ .
  3. Determinamos  $Cs = \{c \in Comp | AH \cap N_{Subgrf} \cap c \neq \emptyset\}$  el conjunto de las componentes conexas de  $Subgrf$  que contengan algún elemento de  $AH \cap N_{Subgrf}$ .
  4. Para cada  $(i, j) \in N$ :
    - a) Si  $(i, j)$  pertenece a un elemento de  $Cs$  entonces:  $Indi_{ij} = 1$  e  $Indacum_{ij} = Indacum_{ij} + 1$ .
    - b)
      - Si  $Indi_{ij} = 0$  entonces  $F_{ij} = u$ .
      - Si  $Indi_{ij} = 1$  y  $Indacum_{ij} \neq 1$  entonces  $F_{ij} = F_{ij}$ .
      - si  $Indi_{ij} = 1$  y  $Indacum_{ij} = 1$  entonces  $F_{ij} = u$ .

## Proceso explicado

Para cada punto  $(i, j)$  va a ocurrir el siguiente proceso a lo largo del bucle. En los primeros valores de  $U$ ,  $Indi_{ij} = Indacum_{ij} = 0$  porque ese nodo del grafo, o bien no está en el conjunto de nivel superior (estricto)  $N_{Subgrf}$  o bien no pertenece a una componente conexa de  $Cs$  (las que tocan los nodos de  $AH$ ). Para esos valores de  $U$  vamos a asignarle  $u$  como valor de la función de filtrado (realmente es una cota superior porque en cada paso la vamos disminuyendo más).

Para un único valor de  $U$ ,  $Indi_{ij} = Indacum_{ij} = 1$ , este valor es el primer umbral en el que  $(i, j) \in A$ , i.e.  $(i, j) \in N_{Subgrf}$  y  $(i, j)$  pertenece a una componente

<sup>3</sup>En nuestra implementación hemos considerado  $p = 0.25$ .

<sup>4</sup>Empezamos los umbrales en 60 porque el valor de DSVa está acotado por  $60kgm^{-2}$  (salvo valores atípicos) y los acabamos en 0 porque DSVa es un variable no negativa.

de las de  $A$ . Después, para los umbrales siguientes, se va a seguir cumpliendo que  $(i, j) \in A$ , ya que va a seguir perteneciendo al conjunto de nivel superior (estricto) y la componente conexas a la que pertenecía no decrece, por tanto,  $Indi_{ij} = 1$  y  $Indacum_{ij} > 1$ , por ello para estos últimos umbrales no vamos a modificar  $F_{ij}$  y nos vamos a quedar con el valor que le asignemos en el primer umbral en el que  $(i, j) \in A$ . En ese primer umbral,  $Indi_{ij} = Indacum_{ij} = 1$ , en el que asignamos a  $F_{ij}$  el umbral actual. Esta asignación es la final, ya que en los umbrales siguientes tenemos que  $Indi_{ij} = 1$  y  $Indacum_{ij} > 1$  y por tanto, no se modifica el valor de  $F_{ij}$ .

### 4.3.2 Ejemplo simple del funcionamiento del algoritmo

En esta subsección vamos a mostrar un ejemplo simple de cómo funciona lo descrito hasta ahora del método. Vamos a suponer que nuestra matriz de datos (que nos da el valor de DSVa a lo largo de la rejilla) es una matriz  $5 \times 5$ . Usamos la matriz de la figura 4.1 (es la misma que la usada en el ejemplo de la subsección 3.3.2).

Primero, debemos aplicar el algoritmo de obtención de función de filtrado (de la subsección 4.3.1) a la matriz. Modelamos entonces la rejilla como un grafo con 25 nodos que están unidos por aristas con los nodos justamente inferiores, superiores, a su izquierda y a su derecha. El grafo que consideramos coincide con el de la figura 3.6. El algoritmo consiste en que se reduce el valor de cada elemento de la matriz hasta que pase por primera vez que, el nodo que representa ese elemento entre en un subgrafo de nivel superior y esté en una componente conexas que toque algunas de las subregiones de interés. Cuando pase eso por primera vez, le asignamos el umbral de dicho subgrafo de nivel superior. Estas están coloreadas de violeta (latitudes de Hawái) y de naranja (continente americano).

Si consideramos un paso de 5 para coincidir con el ejemplo de la subsección 3.3.2, obtenemos los subgrafos de nivel superiores de las figuras 4.2 y 4.3. La matriz resultado de aplicar el algoritmo de obtención de función de filtrado es la matriz de la figura 4.4. En esta matriz resultado hay dos cambios respecto de la inicial:

- Al considerar un paso entre los umbrales de 5, todos los elementos de la matriz se reducen, por lo menos, hasta el múltiplo de 5 justamente inferior

a él. Si considerásemos pasos entre los umbrales más pequeños que 5 este redondeo sería mucho menos brusco.

- Además, los elementos que están en el umbral 35 (y el elemento que está en el 40) no obtienen en la matriz resultado un 35 (40), ya que la componente conexa en la que están no toca ninguna de las subregiones de interés.

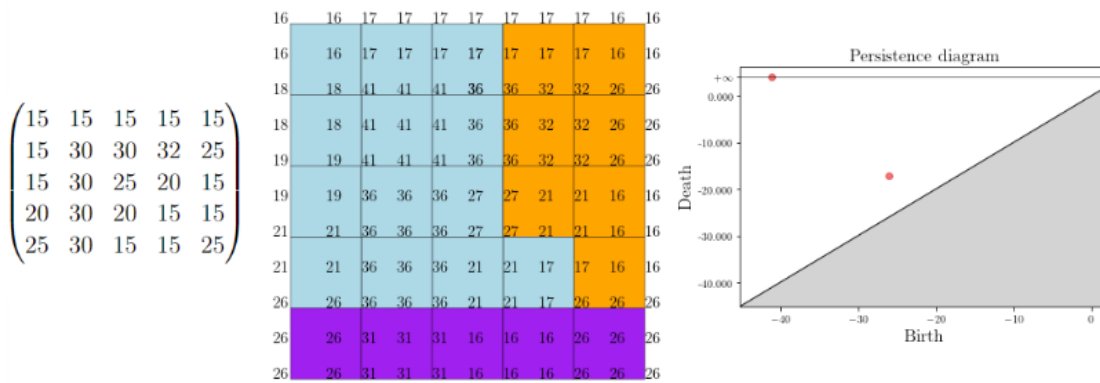


Figura 4.1: A la izquierda vemos una matriz  $5 \times 5$  ejemplo (misma que en la figura 3.6). En el medio vemos su complejo cúbico filtrado asociado. A la derecha vemos el diagrama de persistencia obtenido si aplicásemos homología persistente al complejo cúbico del medio (en color rojo dimensión 0 y en color azul dimensión 1, no hay ninguna clase de homología de dimensión 1). Las coordenadas nacimiento y muerte aparecen multiplicadas por  $-1$ , ya que se ha tenido que invertir la función para usar conjuntos de nivel superior.

Posteriormente, modelamos esta matriz resultado como un complejo cúbico filtrado (podemos verlo en el centro de la figura 4.4) y le aplicamos homología persistente obteniendo la filtración de la figura 4.5 y el consecuente diagrama de persistencia a la derecha en la figura 4.4.<sup>5</sup> Podemos apreciar como aparece una componente conexa en la filtración en el umbral 30, después otra en el umbral 25

<sup>5</sup>En este ejemplo para modelar matriz resultado como un complejo cúbico filtrado y para hacer la filtración hemos usado la segunda solución de la observación 4.1. Sin embargo, al calcular el diagrama de persistencia en el ordenador se tiene que utilizar la primera solución de la observación 4.1 y, por tanto, aparecen las coordenadas de nacimiento y muerte en negativo

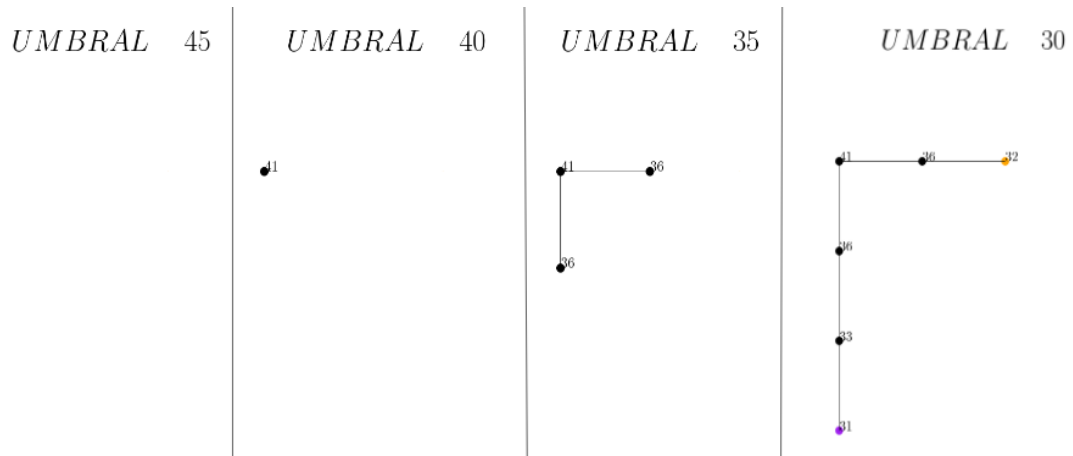


Figura 4.2: Conjuntos de nivel superior del grafo de la figura 3.6 para distintos umbrales.

y finalmente se unen ambas en el umbral 15. Como estamos empleando la *Elder Rule* se considera que la componente que muere es la segunda. Así, el diagrama de persistencia dice que hay una componente conexa que nace en 30 y no muere y otra que nace en 25 y muere en 20. Como no aparecen huecos en la filtración, en el diagrama no aparece ningún punto correspondiente a la dimensión 1.

Finalmente, con fines comparativos, modelamos también la matriz original (fig. 4.1) como un complejo cúbico filtrado y aplicamos homología persistente obteniendo el diagrama de persistencia. Podemos ver el complejo y el diagrama de persistencia en la figura 4.1. La principal diferencia en el diagrama obtenido es que la clase de dimensión 0 (componente conexa) con persistencia infinita nace antes, en el umbral 41. Esto es debido a que en el algoritmo de obtención de filtrado hemos reducido el valor de la función en ciertos puntos para que no naciese esa componente conexa hasta que tocase alguna de las subregiones de interés (ya que una componente conexa que no toque ninguna de las subregiones de interés no forma un patrón de RA).

### 4.3.3 Generación de vectores resultado. Imágenes de persistencia

Una vez hemos modificado las imágenes en escala de grises y después de aplicar homología persistente a las imágenes modificadas, obtenemos un diagrama

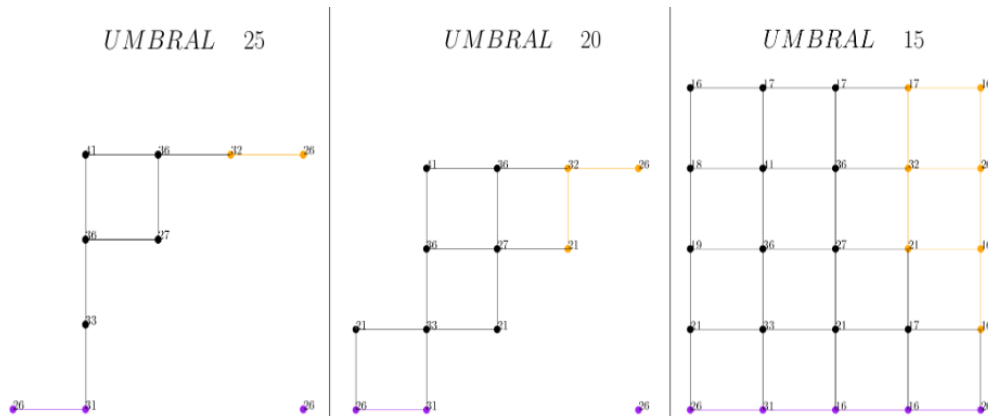


Figura 4.3: Conjuntos de nivel superior del grafo de la figura 3.6 para distintos umbrales.

de persistencia de dimensión 0 y otro de dimensión 1. La pregunta es, ¿cómo obtenemos de esos diagramas un vector que pueda ser *input* de un método de aprendizaje automático posterior?

Para generar un vector *input* de un método de aprendizaje automático vamos a usar imágenes de persistencia. Previamente, en la sección 2.2 las hemos explicado y hemos contado cuáles son los diferentes parámetros que tienen. Las imágenes de persistencia son una forma de medir cuantos puntos (de manera ponderada) hay en cada zona del diagrama de persistencia.

Para cada diagrama de persistencia (tenemos uno asociado a la homología de dimensión 1 y otro a la homología de dimensión 0) realizamos el mismo proceso. Primero transformamos los puntos con multiplicidad positiva y persistencia finita de coordenadas nacimiento-muerte a coordenadas nacimiento-persistencia obteniendo el diagrama transformado. Posteriormente, definimos la superficie de persistencia asociada al diagrama como una suma ponderada de distribuciones de densidad centradas en cada punto del diagrama. Finalmente, escogemos una región de  $\mathbb{R}^2$ , la mínima región que contiene todos los puntos de todos diagramas transformados de la dimensión considerada, 0 o 1 y la dividimos en píxeles de cierto tamaño determinado.

La imagen de persistencia es una imagen en escala de grises formada por los píxeles considerados a lo largo de la región de considerada, donde cada píxel vale el resultado de la integral en el píxel de la superficie de persistencia.

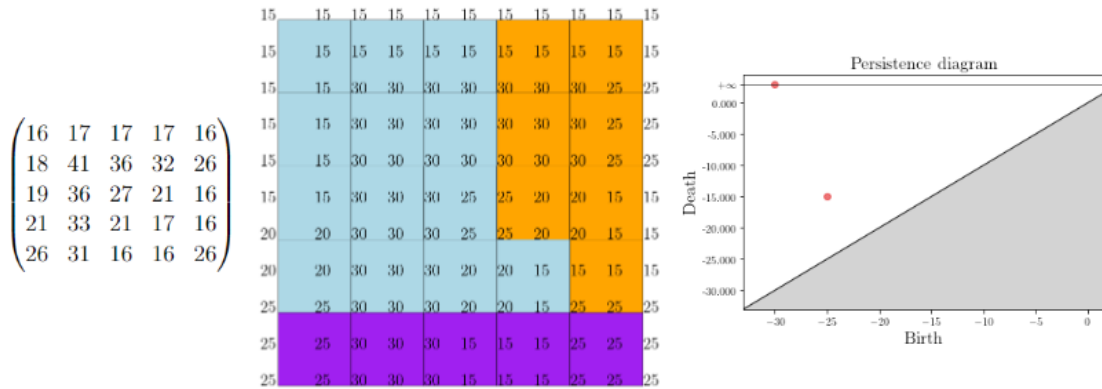


Figura 4.4: A la izquierda vemos la matriz  $5 \times 5$  resultada de aplicarle el algoritmo de obtención de funciones de filtrado de la subsección 4.3.1 a la matriz de la figura 4.1. En el medio vemos el complejo cúbico filtrado asociado a la matriz de la izquierda. A la derecha vemos el diagrama de persistencia obtenido si aplicásemos homología persistente al complejo cúbico del medio (en color rojo dimensión 0 y en color azul dimensión 1, no hay ninguna clase de homología de dimensión 1). Las coordenadas nacimiento y muerte aparecen multiplicadas por  $-1$ , ya que se ha tenido que invertir la función para usar conjuntos de nivel superior. En la figura 4.5 podemos ver la filtración empleada.

Así, debemos escoger tres parámetros. Los escogemos según lo explicado en la sección 2.2:

1. La función de densidad. Escogemos una función de densidad centrada en los puntos y con matriz de varianzas-covarianzas la identidad (i.e., en la notación de la sección 2.2, con  $\sigma = 1$ ).
2. La función de peso. Escogemos la función  $g(x, y) = y$ , es decir, la proyección a la coordenada de la persistencia. Con esta función ponderamos más un punto cuanto más alejado esté de la diagonal, es decir, cuanto sea menos probable que sea ruido topológico (y sea más probable que represente una característica topológica de los datos).
3. Una vez hemos fijado el rango de las imágenes de persistencia para cada dimensión (el rango mínimo que cubra todos los puntos de todos diagramas del conjunto de datos), queda por fijar de qué tamaño son los píxeles para cada dimensión. Hemos escogido que el tamaño de los píxeles de dimensión 0 sea 5 (obteniendo una imagen de persistencia de tamaño  $12 \times 10$ ) y de

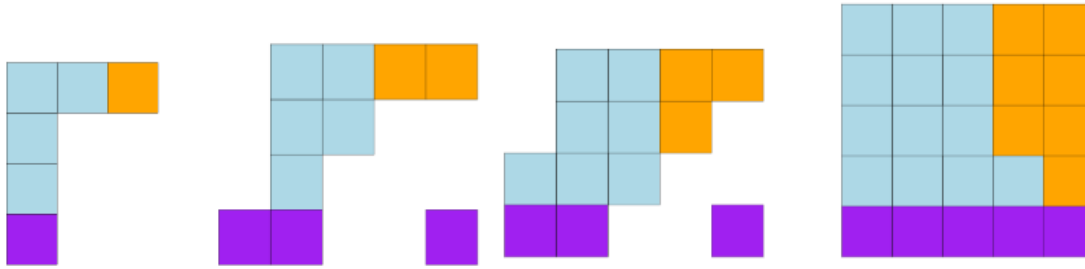


Figura 4.5: Filtración del complejo cúbico filtrado de la figura 4.4, considerando los umbrales (en orden de izquierda a derecha) 30, 25, 20 y 15

dimensión 1 sea 4 (obteniendo una imagen de persistencia de tamaño  $15 \times 9$ ). En la línea de lo comentado en la sección 2.2, queda pendiente comprobar si reduciendo el tamaño la calidad de la clasificación no empeora, lo cual sería una buena noticia porque al reducir el tamaño de las imágenes de persistencia el tiempo de computación del método se reduce.

Una vez obtenemos dos imágenes de persistencia, una de dimensión 1 y otra de dimensión 0, debemos aplanar estas imágenes para formar un único vector. Es decir, debemos pasar de una matriz  $12 \times 10$  a un vector de tamaño 120, donde las primeras 12 coordenadas se corresponden a la primera fila de la matriz, las segundas 12 coordenadas se corresponden a la segunda fila de la matriz, etc.. Del mismo modo, en el caso de dimensión 1, debemos pasar de una matriz  $15 \times 9$  a un vector de tamaño 135. Una vez hemos obtenido un vector por cada dimensión, los empalmamos formando un único vector, que será el *input* de la siguiente sección.

#### 4.3.4 Esquema final y ejemplo de la primera etapa

En la figura 4.6 podemos ver un esquema de toda la etapa del análisis topológico de datos. El esquema representa todo lo que se aplica para cada imagen en escala de grises del conjunto de datos.

Finalmente, mostramos un ejemplo gráfico de la primera etapa empleando una imagen en escala de grises proveniente de los conjuntos de datos de MERRA-2.

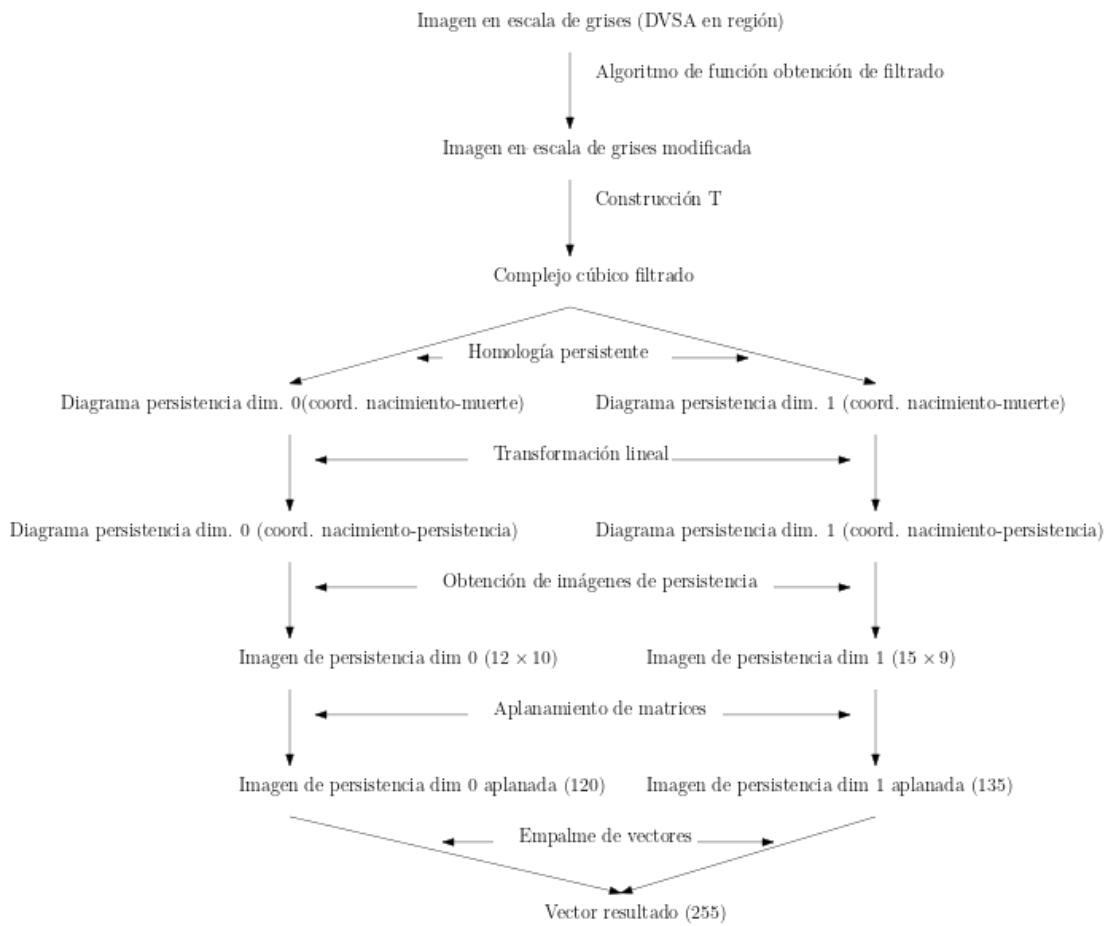


Figura 4.6: Esquema de la etapa 1 del método. El esquema representa todo lo que se aplica para cada imagen en escala de grises del conjunto de datos.



En la imagen mostrada en la figura 4.7 no hay un RA, según las etiquetas provenientes del método TECA-BARD. En la figura 4.7, podemos ver la imagen en escala de grises del ejemplo y a su derecha podemos ver la imagen resultado de aplicarle a la imagen ejemplo el algoritmo para la obtención de una función de filtrado, La principal diferencia entre las dos imágenes es que en la modificada se han reducido los valores máximos, obteniendo un colorado más homogéneo en las zonas donde hay valores medio-altos. Finalmente, debajo de estas dos imágenes en la figura podemos ver los diagramas de persistencia de dimensión 0 y 1 juntos. Estos diagramas son los que se obtienen al aplicarle a la imagen modificada homología persistente.

En la figura 4.8, podemos ver a la izquierda los diagramas de persistencia transformados a coordenadas nacimiento-persistencia (arriba el de dimensión 0, arriba el de dimensión 1). A la derecha de los diagramas de persistencia transformados podemos ver sus imágenes de persistencia asociadas. En ellas se nota que los puntos con más persistencia (los que están más arriba) están mucho más ponderados en la imagen, ya que los píxeles que están más arriba necesitan que haya menos puntos en él (en el diagrama transformados) para tener valores altos.

## 4.4 Segunda etapa, método de los K Vecinos más Cercanos

En esta segunda etapa, una vez obtenidos los vectores resultado de aplicar la etapa anterior a cada imagen en escala de grises del conjunto de datos, debemos clasificar estos vectores según las imágenes de la que provienen se correspondan o no a un río atmosférico. Para ello, vamos a emplear aprendizaje automático y por ello necesitamos usar las etiquetas aportadas por el modelo TECA-BARD (etiquetas binarias que nos indican si las imágenes se corresponden a un RA o no) para entrenar al modelo de aprendizaje automático que empleemos y para medir la calidad de la clasificación que haga el modelo. Por ello, el *input* a este método son: por un lado, los vectores resultado de aplicar la etapa anterior a cada imagen en escala de grises y por otro, las etiquetas binarias resultado de aplicar el modelo TECA-BARD a cada imagen del conjunto de datos.

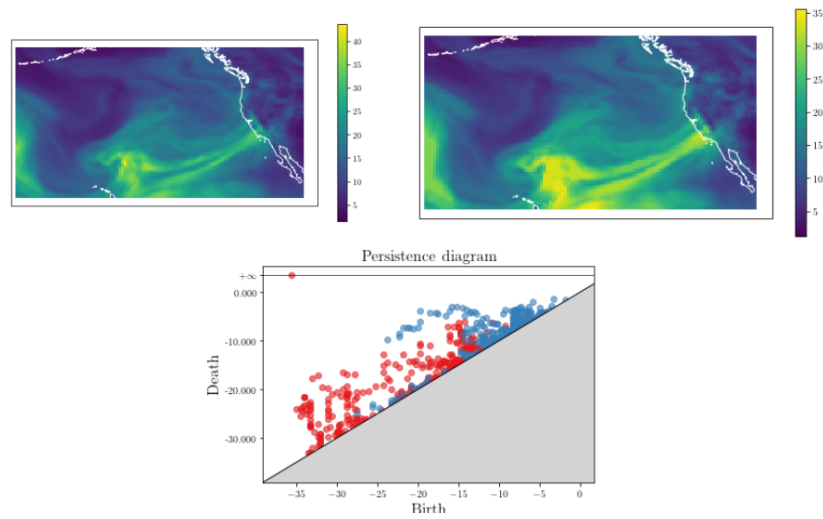


Figura 4.7: A la izquierda podemos ver una imagen en escala de grises donde se mide el valor de DSV en  $kgm^{-2}$ , esta imagen es una de las extraídas de nuestro conjunto de datos salidos del MERRA-2. A la derecha podemos ver la imagen en escala de grises resultado de aplicarle el algoritmo para la obtención de una función de filtrado a la imagen de la izquierda. Abajo, los diagramas de persistencia (dimensión 0 en rojo, dimensión 1 en azul) resultado de aplicarle a la imagen de la izquierda homología persistente en complejos cúbicos.

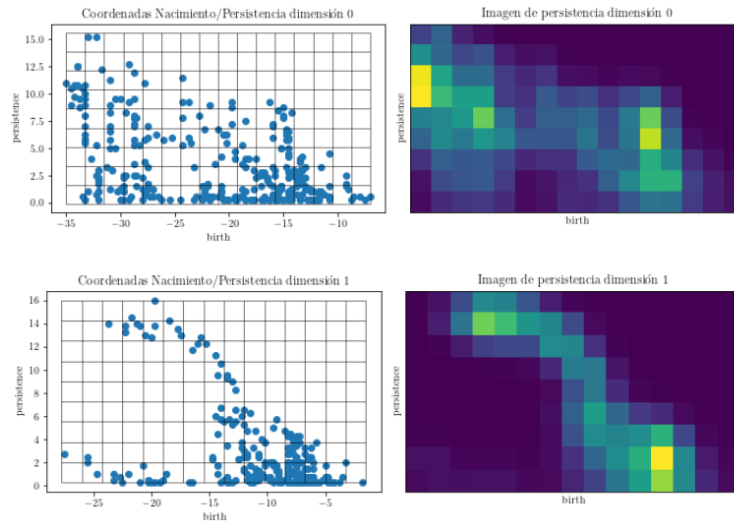


Figura 4.8: Arriba a la izquierda podemos ver el diagrama de persistencia de dimensión 0 de la figura 4.7 transformado a coordenadas nacimiento-persistencia. Arriba a la derecha podemos ver la imagen de persistencia (para este ejemplo hemos considerado excepcionalmente un tamaño de píxel 1.75) del diagrama de dimensión 0 de la figura 4.7. Abajo a la izquierda podemos ver el diagrama de persistencia de dimensión 1 de la figura 4.7 transformado a coordenadas nacimiento-persistencia. Abajo a la derecha podemos ver la imagen de persistencia (para este ejemplo hemos considerado excepcionalmente un tamaño de píxel 1.75) del diagrama de dimensión 1 de la figura 4.7.

Con esos datos se pone en marcha un método de clasificación supervisada donde las variables predictoras son las correspondientes a las características topológicas extraídas previamente y la variable a predecir (a clasificar, en este caso, al ser variable cualitativa) es la variable  $RA$  (valdría 1 si en la observación hay un río atmosférico y  $-1$  en caso contrario). Para más información y detalle sobre los métodos de clasificación supervisada, ver sección 2.3.

Se ha hecho una validación cruzada anidada LTO, procedimiento descrito en la subsección 2.3.2, para determinar cuál es el mejor modelo (entre un conjunto de modelos de aprendizaje automático candidatos) que clasifica mejor estos vectores provenientes de la primera etapa. El mejor modelo es el método de los  $K$  Vecinos Más Cercanos (*K Nearest Neighbours* KNN). El método SVM con núcleo radial (considerado en el método de Muszynski, G.) clasificó ligeramente peor que el KNN (veremos los resultados en la sección 4.6) y no lo hemos considerado.

El KNN predice la clase a la que pertenece un dato del conjunto test como la clase mayoritaria en los  $k$  datos del conjunto de entrenamiento más cercanos (en el espacio dónde están las variables predictoras). Para más detalles sobre el KNN consultar la sección 2.10.

El KNN depende de dos opciones. La primera es la distancia escogida para medir qué observaciones del conjunto de entrenamiento están más cerca. Hemos utilizado la distancia euclidiana. La segunda es escoger  $K$ , el número de vecinos considerados en el conjunto de entrenamiento. Para escoger este valor de  $K$ , entre un conjunto de valores candidatos, hemos empleado validación cruzada (no anidada). Este procedimiento y sus motivaciones están descritos en la sección 2.3.1. La importancia particular del valor de  $K$  está explicada en la sección 2.10.

Una vez entrenado el KNN y habiendo escogido el  $K$  óptimo, predecimos los valores para el conjunto test y calculamos las medidas de calidad de clasificación (descritas en 2.3.3).

*Observación 4.2.* Al igual que otros métodos de aprendizaje automático, el KNN, tiene a sobreajustarse a la clase mayoritaria, por tanto, tiende a clasificar bastante bien los datos de la clase mayoritaria y mal los datos de la clase minoritaria. Para resolver este problema se ha aplicado remuestreo para obtener conjuntos de entrenamiento con clases equilibradas. El remuestreo se ha realizado antes de la división en conjunto de entrenamiento y test. Este remuestreo también se ha hecho en la validación cruzada anidada LTO previa en la que hemos seleccionado el mejor método de aprendizaje automático.

## 4.5 Implementación y código

Hemos programado cada etapa de este método en dos lenguajes distintos. El código de implementación está disponible en:

<https://github.com/davafe/Metodo-deteccion-rios-atmosfericos>.

La primera etapa se ha programado en Python. Para leer los archivos *netCDF* con los datos obtenidos de MERRA-2 se ha empleado la librería *netCDF4* ([26]). Hemos programado el algoritmo de obtención de función de filtrado (el algoritmo de la sección 4.3.1) y hemos utilizado el paquete *cv2* ([27]) para calcular las componentes conexas de los subgrafos de nivel. En este algoritmo hemos considerado un paso entre umbrales de  $p = 0.25$ . Para modelar las imágenes en escala de grises como complejos cúbicos y computar la homología persistente de esos complejos hemos empleado el módulo *Cubical Complex* de **gudhi** ([28]). Finalmente, para calcular y representar las imágenes de persistencia hemos usado el módulo *Persistence Imager* de *persim* ([29]).

La segunda etapa se ha programado en R. Destacar que hemos empleado la implementación de KNN del paquete *class* ([30]).

El tiempo de computación de la primera etapa con paso  $p = 0.25$  fue de alrededor de 30 horas. Con paso  $p = 1$  fue de 8 horas aproximadamente. Por lo tanto, el tiempo de computación depende mucho del paso escogido. En la segunda etapa, la computación duró alrededor de 1 hora. Estos tiempos de computación se alcanzaron en un ordenador personal con procesador Intel Core *i5*.

## 4.6 Resultados

Una vez aplicado el KNN a los vectores surgidos de aplanar y juntar las imágenes de persistencia, se mide la calidad de su clasificación en tres de las medidas de bondad que se han definido en la sección 2.3.3: exactitud, precisión y sensibilidad. Estas tres medidas se miden (valga la redundancia) en el conjunto test.

En el cuadro 4.2 podemos ver la exactitud para los dos conjuntos de datos empleados (realmente son el mismo solo que en uno consideramos imágenes cada 3 horas y en otro solo 1 de esas 8 imágenes diarias). Podemos ver como con datos

Cuadro 4.2: Tabla con la exactitud en el conjunto test del KNN, para los distintos conjuntos de datos considerados.

Mod. climático	Resol. temporal	Resol. espacial	Exactitud
MERRA-2	3h	50 km	0.8324
MERRA-2	24h	50 km	0.7209

Cuadro 4.3: Tabla con la precisión y sensibilidad en el conjunto test del KNN, para los distintos conjuntos de datos considerados.

Mod. climático	Resol. temporal	Resol. espacial	Precisión	Sensibilidad
MERRA-2	3h	50 km	0.7816	0.9026
MERRA-2	24h	50 km	0.7171	0.6728

cada 3 horas obtenemos un resultado bastante bueno, con más del 83 % de imágenes correctamente. Sin embargo, si reducimos la precisión temporal, la exactitud se reduce bastante, más de un 10 %, obteniendo una exactitud mejorable.

En el cuadro 4.3 podemos ver para los dos conjuntos de datos empleados la precisión y sensibilidad. Con precisión horaria de 3h obtenemos una magnífica sensibilidad, del 0.9026, es decir, que más del 90 % de los ríos atmosféricos se detectan. En este mismo conjunto de datos obtenemos una precisión no tan buena, del 0.7816, i.e., de las imágenes clasificadas como ríos atmosféricos, el 78.16 % son realmente ríos atmosféricos. Si reducimos la precisión horaria, la sensibilidad baja mucho, más de un 22 % y la precisión no baja tanto, algo más del 6,4 %.

#### 4.6.1 Importancia de las dimensiones empleadas en homología y del método de aprendizaje automático escogido

Una vez vistos estos resultados generales, cabe preguntarse dos cosas sobre la importancia de las novedades en nuestro método:

- ¿Si solo calculásemos la homología en dimensión 0, como en el método anterior, obtendríamos el mismo resultado? Dicho de otra manera, ¿incluir

Cuadro 4.4: Tabla con las distintas exactitudes, precisiones y sensibilidades obtenidas por dos distintos métodos (KNN y SVM con núcleo radial) y empleando las imágenes de persistencia de dimensiones 0 y/o 1.

Método	Dimensiones empleadas	Exactitud	Precisión	Sensibilidad
KNN	0,1	0.8324	0.7816	0.9022
KNN	0	0.8091	0.7643	0.8693
KNN	1	0.7717	0.7109	0.8811
SVM Radial	0,1	0.8310	0.8183	0.8320
SVM Radial	0	0.8162	0.7970	0.8264
SVM Radial	1	0.7183	0.6956	0.7334

la información topológica sobre homología en dimensión 1 realmente mejora la clasificación?

- ¿Realmente utilizar KNN es mejor que emplear SVM con núcleo radial?

Hemos computado nuestro método empleando en la clasificación solo imágenes de persistencia de dimensión 0 y solo utilizando las imágenes de persistencia de dimensión 1. Además, hemos probado el método de aprendizaje automático SVM radial (explicado en la sección 2.4) sobre imágenes de dimensión 0 y/o 1. La calidad de las clasificaciones de cada combinación de métodos y dimensiones empleadas se puede ver en el cuadro 4.4 en términos exactitud, precisión y sensibilidad.

En primer lugar, fijándonos solamente en el método KNN, vemos como usar homología persistente en dimensión 0 y 1 mejora en más de un 2,2% la exactitud, en más de un 1,5% la exactitud y en más de un 4% la sensibilidad respecto a si solamente utilizásemos homología persistente en dimensión 0. Además, empleando homología persistente solo en dimensión 1 obtenemos una buena exactitud, 0.7717, que si bien no mejora la también buena exactitud con solo dimensión 0, sí que obtenemos mejor sensibilidad con solamente dimensión 1 que con solamente dimensión 0. Resumidamente, la homología persistente en dimensión 1 aporta una mejora en la clasificación, especialmente en términos de sensibilidad (i.e., en clasificar correctamente las imágenes en escala de grises correspondientes a ríos atmosféricos)

En segundo lugar, haciendo la comparativa KNN contra SVM núcleo radial,

podemos ver que KNN obtiene una exactitud ligeramente superior, empleando ambas dimensiones de homología, y una bastante superior, utilizando solo la dimensión 1 de homología, que la que se obtiene con SVM radial. Sin embargo, si solo se usase la dimensión 0 la SVM obtiene una exactitud algo superior. En términos de sensibilidad, el KNN es mejor (independientemente de que dimensiones se usen), sin embargo, la SVM es mejor en términos de exactitud (salvo si se emplea solamente dimensión 1). Concluimos que, de manera general, KNN obtiene mejores resultados que SVM, salvo que se use solo la dimensión 0.

Finalmente, recordar que el modelo de aprendizaje automático está sesgado por las etiquetas empleadas provenientes de TECA-BARD, ya que hemos asumido que estas etiquetas son "verdad" y hemos entrenado y calificado nuestro modelo con ellas. Queda pendiente de comprobar y usar el método con distintos etiquetados provenientes de otros métodos.



# 5 | Comparación de métodos: Estabilidad y resultados numéricos

En este capítulo compararemos el método de Muszynski, G. et al. con el propuesto en este trabajo en términos de estabilidad y de resultados numéricos. En la sección 5.1 demostramos la estabilidad de nuestro método con un paso suficientemente pequeño y probamos la no estabilidad del método de Muszynski, G. et al. aunque el paso tienda a 0. En la sección 5.2 comparamos los resultados empíricos de ambos métodos.

## 5.1 Estabilidad

La estabilidad es una propiedad importante que nos indica si el resultado de un algoritmo o método cambia mucho si el *input* cambia ligeramente. Que un método sea estable nos garantiza que una pequeña alteración en el *input* no va a provocar grandes cambios en el resultado.

En esta sección compararemos la estabilidad que tienen la primera etapa, en la que se hace ATD, el método de Muszynski, G. et al. (descrito en el capítulo 3) y el método propuesto en este trabajo (descrito en el capítulo 4).

Como ambos métodos tienen como *input* unas imágenes en escala de grises (que son matrices) y como salida un vector de  $\mathbb{R}^k$ , recordamos las definiciones de las normas de matrices y vectores que vamos a usar en esta sección.

**| Definición 5.1.** Sea  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , la norma uno y la norma infinito de  $x$  se definen, respectivamente:

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad \|x\|_\infty = \max_{i \in \{1, 2, \dots, n\}} |x_i|.$$

Sea  $A = (A_{ij}) \in M_{n,m}(\mathbb{R})$ , la norma uno y la norma infinito de  $A$  se definen, respectivamente:

$$\|A\|_1 = \sum_{i=1}^n \sum_{j=1}^m |a_{ij}|, \quad \|A\|_\infty = \max_{i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}} |a_{ij}|.$$

Estas dos normas (tanto en el caso de matrices como en el de vectores) son equivalentes, es decir, la topología que inducen es la misma, ya que, claramente, se cumple que:

$$\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty \quad \forall x \in \mathbb{R}^n \quad (5.1)$$

$$\|A\|_\infty \leq \|A\|_1 \leq nm\|A\|_\infty \quad \forall A \in M_{n,m}(\mathbb{R}) \quad (5.2)$$

En esta sección, vamos a probar la estabilidad en norma infinito, sin embargo, por esta equivalencia, sería análogo hacerlo en norma uno.

Como, tanto el método de Muszynski, G. et al. y el nuestro, dependen de un paso  $p > 0$  entre umbrales considerados (ya que se necesita considerar un conjunto finito de umbrales), vamos a comparar la estabilidad cuando ese paso tiende a cero. Es decir, se compara la estabilidad de la versión del método considerando todos los umbrales en un intervalo. También vamos a estudiar la estabilidad cuando el paso es menor que la precisión de los datos, es decir, cuando tenemos la suficiente potencia computacional para considerar todos los distintos subconjuntos posibles de nivel superior.

Definimos ahora lo que consideramos como estabilidad:

**| Definición 5.2.** Sea  $f: M_{n,m}(\mathbb{R}) \rightarrow \mathbb{R}^k$  una función. Se dice que  $f$  es estable si existe una constante  $C \geq 0$  tal que:

$$\|f_A - f_B\|_\infty \leq C\|A - B\|_\infty \quad \forall A, B \in M_{n,m}(\mathbb{R}).$$

Definimos la estabilidad de un método que tiene como input elementos de  $M_{n,m}(\mathbb{R})$  y output elementos de  $\mathbb{R}^k$ , como la estabilidad de la función:

$$f: A \in M_{n,m}(\mathbb{R}) \mapsto f_A \in \mathbb{R}^k$$

donde  $f_A$  es el output del método si el input es  $A$ .

Así, si un método es estable, la norma de la diferencia de los *outputs* está acotada por la norma de la diferencia de los *inputs* multiplicada por una constante. Claramente que una  $f$  sea estable implica que  $f$  sea continua (considerando las topologías inducidas por la norma).

En la subsección 5.1.1 demostraremos que cuando  $p \rightarrow 0$  la primera etapa del método que hemos propuesto en el capítulo 4 es estable y, finalmente, demostraremos que no es necesario que  $p \rightarrow 0$ , sino que el paso sea menor que la precisión de los datos, para que nuestro método sea estable. En la subsección 5.1.2 demostraremos que la primera etapa del método de Muszynski, G. et al. (descrito en el capítulo 3) no cumple estas dos propiedades. Así, en términos de estabilidad nuestro método es mejor que el de Muszynski, G. et al..

### 5.1.1 Estabilidad de la primera etapa del método propuesto

En esta subsección vamos a demostrar la estabilidad (cuando el paso tiende a 0) de la primera etapa de nuestro método. Dividiremos esta demostración en dos partes: primero demostraremos la estabilidad (cuando el paso tiende a 0) del algoritmo de función de filtrado y después, empleando la estabilidad de los diagramas de persistencia y de las imágenes de persistencia, concluimos la estabilidad (cuando el paso tiende a 0) de toda la etapa de ATD de nuestro método. Finalmente, veremos que fijando un paso menor que la precisión de los datos obtenemos estabilidad.

#### Estabilidad del algoritmo de función de filtrado

Introducimos primero la notación que vamos a utilizar. Después, demostraremos 3 lemas sobre el algoritmo de función de filtrado, que finalmente, usaremos para probar la estabilidad del algoritmo (cuando el paso tiende a 0).

Sea  $N = \{(i, j) | i \in \{1, \dots, n\}, j \in \{1, \dots, m\}\}$  los nodos de la rejilla subyacente al grafo  $G$ , donde  $n, m \in \mathbb{N}$  son el número de latitudes y longitudes (respectivamente) diferentes que tienen los píxeles de la rejilla. Sea  $f$  la siguiente función:

$$f: A \in M_{n,m}(\mathbb{R}) \mapsto f_A \in M_{n,m}(\mathbb{R})$$

dónde  $f_A$  es la matriz  $F$  resultante de aplicar el algoritmo de la página 65 con *input*  $A$ . Dado  $A \in M_{n,m}(\mathbb{R})$ , denotamos el elemento en la posición  $(i, j)$  por  $A(i, j)$ .

Dado un  $\lambda \geq 0$ , denotamos  $t(\lambda) := \min\{u \in U | u \geq \lambda\}$  el umbral (elemento de  $U$ ) inmediatamente superior a  $\lambda$  y  $s(\lambda) := \max\{u \in U | u \leq \lambda\}$  el umbral (elemento de  $U$ ) inmediatamente inferior a  $\lambda$ . El resto de la notación es la misma que en 4.3.1.

Una vez introducida la notación, demostraremos tres lemas cuasi-inmediatos que se utilizarán para probar la estabilidad (cuando el paso tiende a cero). En los tres lemas se usa la idea de que  $f_A(i, j)$  vale el primer umbral tal que para el cual ese punto de la rejilla entra en el conjunto de nivel superior (estricto) en una componente conexa que toque algún punto correspondiente con América o las latitudes de Hawái.

**Lema 5.1.** Para todo  $(i, j) \in N$   $f_A(i, j) \in U = \{60, 60 - p, 60 - 2p, \dots, 0\}$ .

*Demostración.* Por definición del algoritmo de obtención de función de filtrado, la matriz  $f_A$  solo toma valores umbrales. |

**Lema 5.2.** Dado  $(i, j) \in N$   $f_A(i, j) \geq \lambda$  si, y solo si, existe un camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ , tal que  $A(l, k) > t(\lambda) \quad \forall (l, k) \in C$ , siendo  $C \subset N$  el conjunto formado los nodos de dicho camino.

*Demostración.* Sea  $(i, j) \in N$ . Supongamos que  $f_A(i, j) \geq \lambda$ , entonces, tal y como está formulado el algoritmo, existe un  $u \in U$ ,  $u \geq \lambda$ , tal que en el subconjunto de nivel superior (estricto) correspondiente a  $u$  el punto  $(i, j)$  pertenece a la misma componente conexa que un elemento de  $AH$ . Por tanto, existe un camino entre  $(i, j)$  y un punto de  $AH$  tal que  $A(k, l) > u$  para todo  $(l, k) \in C \subset N$ , siendo  $C$  el conjunto de nodos de dicho camino. Como  $u \in \{v \in U | v \geq \lambda\}$ , entonces  $u \geq t(\lambda)$  y  $A(k, l) > t(\lambda)$  para todo  $(l, k) \in C \subset N$ .

Supongamos ahora que existe un camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ , tal que  $A(l, k) > t(\lambda) \quad \forall (l, k) \in C'$ , siendo  $C' \subset N$  el conjunto formado los nodos de dicho camino. Por lo tanto, como  $\lambda \leq t(\lambda) \in U$ , existe un umbral

no inferior a  $\lambda$  tal que el subconjunto de nivel superior (estricto) correspondiente a  $u$  el punto  $(i, j)$  pertenece a la misma componente conexa que un elemento de  $AH$  y, por ende,  $f_A(i, j) \geq \lambda$  |

**Lema 5.3.** Dado  $(i, j) \in N$   $f_A(i, j) \leq \lambda$  si, y solo sí, no existe ningún camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ , tal que  $A(l, k) > s(\lambda) + p \quad \forall (l, k) \in C$ , siendo  $C \subset N$  el conjunto formado los nodos de dicho camino.

**Demostración.** Sea  $(i, j) \in N$ . Supongamos que  $f_A(i, j) \leq \lambda$ , entonces, tal y como está formulado el algoritmo, no existe un  $u \in U$ ,  $u > \lambda$ , tal que en el subconjunto de nivel superior (estricto) correspondiente a  $u$  el punto  $(i, j)$  pertenece a la misma componente conexa que un elemento de  $AH$ . Por tanto, no existe un camino entre  $(i, j)$  y un punto de  $AH$  tal que  $A(k, l) > u$  para todo  $(l, k) \in C \subset N$ ,  $u \in U$ ,  $u > \lambda$ , siendo  $C$  el conjunto de nodos de dicho camino. Como  $s(\lambda) + p > \lambda$  y  $\lambda \in U$ , entonces concluimos que se cumple el segundo enunciado de la equivalencia a demostrar.

Supongamos que no existe ningún camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ , tal que  $A(l, k) > s(\lambda) + p \quad \forall (l, k) \in C'$ , siendo  $C' \subset N$  el conjunto formado los nodos de dicho camino. Tenemos dos casos:

- Si  $\lambda \notin U$ , entonces  $s(\lambda) + p = \lambda$  y en consecuencia, no existe un umbral mayor o igual a  $\lambda$  tal que  $(i, j)$  pertenezca a la misma componente conexa en el conjunto de nivel superior (estricto) que un elemento de  $AH$ , por tanto, tal y como está descrito el algoritmo,  $f_A(i, j) < \lambda$ .
- Si  $\lambda \in U$ , entonces  $\lambda = s(\lambda)$  consideramos el caso anterior sustituyendo  $\lambda$  por  $s(\lambda) + p = \lambda + p$ . Por lo tanto,  $f_A(i, j) < \lambda + p$ , y, por el lema 5.1,  $f_A(i, j) \leq \lambda$ , ya que el umbral justo inferior a  $\lambda + p$  es  $\lambda$ .

|

Con estos lemas demostrados, ya somos capaces de demostrar la estabilidad del algoritmo de obtención de función de filtrado, cuando el paso tiende a 0:

**| Teorema 5.1 (Estabilidad, con  $p \rightarrow 0$ , del algoritmo).** Sean  $A, B \in M_{n,m}(\mathbb{R})$  y un paso  $p > 0$ , se cumple que:

$$\|f_A - f_B\|_\infty \leq \|A - B\|_\infty + p$$

**Demostración.** Sea  $(i, j) \in N$ , sea  $\delta = \|A - B\|_\infty$  y sea  $\lambda = f_A(i, j)$ . Así,  $\lambda \in U$  (lema 5.1) y por tanto,  $\lambda = t(\lambda) = s(\lambda)$

Veamos primero que  $f_B(i, j) \geq t(\lambda - \delta - p) = t(f_A(i, j) - \delta - p)$ :

Como  $\lambda = f_A(i, j)$ , por el lema 5.2, existe un camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ , siendo  $C \subset N$  el conjunto formado por los nodos de dicho camino, tal que  $A(l, k) > t(\lambda) = \lambda \quad \forall (l, k) \in C$ .

Por la definición de la norma infinito,  $|A(l, k) - B(l, k)| \leq \delta \quad \forall (l, k) \in C$ . Por tanto:

$$B(l, k) \geq A(l, k) - \delta > \lambda - \delta \geq t(\lambda - \delta - p) \quad \forall (l, k) \in C$$

y por el lema 5.2 tenemos que  $f_B(i, j) \geq t(\lambda - \delta - p)$ .

Veamos ahora que  $f_B(i, j) \leq \lambda + t(\delta) = f_A(i, j) + t(\delta)$ :

Supongamos que  $f_B(i, j) > \lambda + t(\delta)$ , entonces, por el lema 5.3:

$$B(l, k) > s(\lambda + t(\delta)) + p = \lambda + s(t(\delta)) + p \quad \forall (k, l) \in C,$$

siendo  $C'$  los nodos de un camino entre algún punto de  $AH$  e  $(i, j)$  en el grafo  $G$ . Entonces, como  $|A(l, k) - B(l, k)| \leq \delta \quad \forall (l, k) \in C'$ , tenemos que:

$$\lambda + t(\delta) + p = \lambda + s(t(\delta)) + p < B(l, k) \leq A(l, k) + \delta \quad \forall (l, k) \in C'$$

Así,  $A(l, k) > \lambda + p + t(\delta) - \delta \geq \lambda + p = s(\lambda) + p \quad \forall (l, k) \in C'$ , y por tanto, llegamos a una contradicción (usando lema 5.3) con que  $f_A(i, j) = \lambda$ . Por ello concluimos que  $f_B(i, j) \leq \lambda + t(\delta)$ .

Por tanto, hemos demostrado que, para todo  $(i, j) \in N$ , se tiene:

$$f_A(i, j) - \delta - p \leq t(f_A(i, j) - \delta - p) \leq f_B(i, j) \leq f_A(i, j) + t(\delta) \leq f_A(i, j) + \delta + p,$$

y, por tanto,  $|f_A(i, j) - f_B(i, j)| \leq \delta + p = \|A - B\|_\infty + p$ .

Finalmente, la definición de norma infinito implica que  $\|f_A - f_B\|_\infty \leq \|A - B\|_\infty + p$ . █

### Estabilidad de la primera etapa del método propuesto

Ahora, empleando la estabilidad de los diagramas de persistencia y de las imágenes de persistencia vamos a demostrar la estabilidad (cuando  $p \rightarrow 0$ ) de la primera etapa del método propuesto. Después vamos a demostrar la estabilidad cuando el paso es menor que la precisión de los datos. Para ello, primero tenemos que definir una distancia en los diagramas de persistencia.

**| Definición 5.3.** La distancia  $1, \infty$  de Wasserstein entre dos diagramas de persistencia  $B$  y  $B'$  está dada por:

$$W_{1,\infty}(B, B') = \inf_{\gamma: B \rightarrow B'} \left( \sum_{u \in B} \|u - \phi(u)\|_{\infty} \right)$$

y la distancia  $1, 1$  de Wasserstein entre dos diagramas de persistencia  $B$  y  $B'$  está dada por:

$$W_{1,1}(B, B') = \inf_{\gamma: B \rightarrow B'} \left( \sum_{u \in B} \|u - \phi(u)\|_1 \right),$$

donde  $\gamma$  recorre el conjunto de las biyecciones entre  $B$  y  $B'$ .

Como el ínfimo conserva el orden, la desigualdad (5.1) implica que:

$$W_{1,\infty}(B, B') \leq W_{1,1}(B, B') \text{ para todo } B, B' \text{ diagramas de persistencia} \quad (5.3)$$

**| Teorema 5.2.** Sean  $A, B \in M_{n,m}(\mathbb{R})$  dos imágenes en escala de grises de dimensión 2 y sean  $Dgm_k(A)$  y  $Dgm_k(B)$  dos diagramas de persistencia de dimensión  $k \in \{0, 1, \dots\}$  asociados a aplicar homología persistente, con filtraciones de subcomplejos de nivel inferior, a los complejos cúbicos filtrados asociados empleando la construcción  $T$  a las imágenes en escala de grises  $A$  y  $B$ , respectivamente. Se tiene que:

$$W_{1,1}(Dgm_k(A), Dgm_k(B)) \leq 9\|A - B\|_1$$

Este teorema fue demostrado por P. Skraba y K Turner en [31]. Hemos explicado la homología persistente y los complejos cúbicos filtrados en la sección 2.1.

Tal y como hemos explicado en la observación 4.1, considerar filtraciones de nivel superior en un complejo cúbico filtrado a partir de una imagen en escala de grises es equivalente a considerar las filtraciones de nivel inferior pero a partir de la imagen cambiada de signo. De esto y de que  $\|A - B\|_1 = \|(-A) - (-B)\|_1$  obtenemos el teorema análogo para filtraciones de subcomplejos de nivel superior.

**| Teorema 5.3.** Sean  $A, B \in M_{n,m}(\mathbb{R})$  dos imágenes en escala de grises de dimensión 2 y sean  $Dgm_k(A)$  y  $Dgm_k(B)$  dos diagramas de persistencia de dimensión  $k \in \{0, 1, \dots\}$  asociados a aplicar homología persistente, con filtraciones de subcomplejos de nivel superior, a los complejos cúbicos filtrados asociados

empleando la construcción  $T$  a las imágenes en escala de grises  $A$  y  $B$ , respectivamente. Se tiene que:

$$W_{1,1}(Dgm_k(A), Dgm_k(B)) \leq 9\|A - B\|_1.$$

Este teorema es el que vamos a emplear en la demostración, ya que en nuestro método hemos usado subcomplejos de nivel superior.

También vamos a usar el teorema siguiente sobre estabilidad de las imágenes de persistencia (explicadas en la sección 2.2):

**| Teorema 5.4.** Sean  $B$  y  $B'$  dos diagramas de persistencia y  $I(\rho_B)$  y  $I(\rho_{B'})$  las imágenes de persistencia asociadas a  $B$  y  $B'$ , respectivamente, con función de distribución normal bivalente (con matriz de covarianzas diagonal y las varianzas de las dos variables valen  $\sigma^2$ ) y una función  $g$  de ponderación no negativa, continua, diferenciable a trozos y que vale cero en el eje horizontal del plano (si la persistencia es nula). Se tiene que:

$$\|I(\rho_B) - I(\rho_{B'})\|_\infty \leq \left(5|\nabla g| + \sqrt{\frac{10}{\pi}} \frac{\|g\|_\infty}{\sigma}\right) W_{1,\infty}(B, B')$$

donde  $\|g\|_\infty = \sup g(x, y)$  y  $|\nabla g| = \sup \|\nabla g\|_2$

Este teorema fue demostrado por Adams, H., Emerson, T en [15].

Con todos estos resultados podemos demostrar la estabilidad, cuando el paso tiende a 0, de la primera etapa de nuestro método:

**| Teorema 5.5 (Estabilidad (con  $p \rightarrow 0$ ) del método propuesto).** Sean  $A, B \in M_{n,m}(\mathbb{R})$  dos imágenes en escala de grises,  $p > 0$  un paso y  $h: C \in M_{n,m}(\mathbb{R}) \mapsto h_C \in \mathbb{R}^k$ , donde  $h_C$  es el resultado de aplicarle a la matriz  $C$  la primera etapa de nuestro método, la descrita en la sección 4.3. Se tiene que

$$\|h_A - h_B\|_\infty \leq \left( \left(5 + \sqrt{\frac{10}{\pi}} 60\right) 9nm(\|A - B\|_\infty + p) \right).$$

*Demostración.* Sean  $A, B \in M_{n,m}(\mathbb{R})$  y  $h_A, h_B \in \mathbb{R}^k$  los resultados de aplicarle a las matrices  $A$  y  $B$ , respectivamente, el método descrito en la sección 4.3. Veamos paso por paso, de principio a final, todo el proceso que se les ha aplicado a las matrices y así vamos acotando en cada paso hasta acotar  $\|h_A - h_B\|_\infty$ .



1. Lo primero que se les aplica a  $A$  y  $B$  es el *algoritmo de obtención de función de filtrado* (sec. 4.3.1) y, por el teorema 5.1:

$$\|f_A - f_B\|_\infty \leq \|A - B\|_\infty + p \quad (5.4)$$

dónde  $f_A$  y  $f_B$  son las matrices resultantes de aplicar dicho algoritmo a  $A$  y  $B$ , respectivamente.

2. De  $f_A$  y  $f_B$  se obtienen  $Dgm_k(A)$  y  $Dgm_k(B)$  con  $k = 0$  y  $k = 1$ .  $Dgm_k(A)$  y  $Dgm_k(B)$  son dos diagramas de persistencia de dimensión  $k \in \{0, 1, \dots\}$  asociados a aplicar homología persistente, con filtraciones de subcomplejos de nivel superior, a los complejos cúbicos filtrados asociados empleando la construcción T a las imágenes en escala de grises  $f_A$  y  $f_B$ . Podemos aplicar entonces el teorema 5.3 y obtener:

$$W_{1,1}(Dgm_0(f_A), Dgm_0(f_B)) \leq 9\|f_A - f_B\|_1.$$

$$W_{1,1}(Dgm_1(f_A), Dgm_1(f_B)) \leq 9\|f_A - f_B\|_1.$$

Aplicando las desigualdades (5.2) y (5.4) obtenemos que:

$$W_{1,1}(Dgm_0(f_A), Dgm_0(f_B)) \leq 9nm\|f_A - f_B\|_\infty \leq 9nm(\|A - B\|_\infty + p) \quad (5.5)$$

$$W_{1,1}(Dgm_1(f_A), Dgm_1(f_B)) \leq 9nm\|f_A - f_B\|_\infty \leq 9nm(\|A - B\|_\infty + p) \quad (5.6)$$

3. De  $D_A^0 := Dgm_0(f_A)$ ,  $D_B^0 := Dgm_0(f_B)$  se obtienen  $I(\rho_{D_A^0})$  y  $I(\rho_{D_B^0})$ .  $I(\rho_{D_A^0})$  y  $I(\rho_{D_B^0})$  son resultado de calcular las imágenes de persistencia a los diagramas de persistencia  $D_A^0, D_B^0$  en las condiciones del teorema 5.4, con  $\sigma = 1$  y  $g(x, y) = y$ , por tanto, por dicho teorema obtenemos que:

$$\|I(\rho_{D_A^0}) - I(\rho_{D_B^0})\|_\infty \leq \left(5|\nabla g| + \sqrt{\frac{10}{\pi}}\|g\|_\infty\right)W_{1,\infty}(D_A^0, D_B^0).$$

De manera análoga, para  $D_A^1 := Dgm_1(f_A)$ ,  $D_B^1 := Dgm_1(f_B)$  se obtienen  $I(\rho_{D_A^1})$  y  $I(\rho_{D_B^1})$  cumpliendo que:

$$\|I(\rho_{D_A^1}) - I(\rho_{D_B^1})\|_\infty \leq \left(5|\nabla g| + \sqrt{\frac{10}{\pi}}\|g\|_\infty\right)W_{1,\infty}(D_A^1, D_B^1).$$

Como cualquier matriz resultante del algoritmo de función de filtrado va a tomar valores en el conjunto de umbrales  $U = \{60, 60-p, 60-2p, \dots, 0\}$  (por el lema 5.1), todo diagrama de persistencia que obtengamos en el segundo paso va a obtener entre 0 y 60. Además, en las imágenes de persistencia (ver

la sección 2.2) solo se emplea la función de ponderación  $g$  sobre los diagramas en coordenadas nacimiento persistencia. Por tanto, podemos restringir la función  $g$  a puntos de  $\mathbb{R}^2$  con la segunda coordenada, la de persistencia, entre 0 y 60. Además, como en este trabajo consideramos  $g(x, y) = y$  (la proyección a la coordenada de persistencia), podemos acotar  $g$ , obteniendo  $\|g\|_\infty \leq 60$ .

Por otra parte, como  $g(x, y) = y$ ,  $\nabla g(x, y) = (0, 1)$  para todo  $(x, y)$ , por lo tanto,  $|\nabla g| = 1$ .

Con estas dos consideraciones obtenemos que:

$$\|I(\rho_{D_A^0}) - I(\rho_{D_B^0})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)W_{1,\infty}(D_A^0, D_B^0),$$

$$\|I(\rho_{D_A^1}) - I(\rho_{D_B^1})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)W_{1,\infty}(D_A^1, D_B^1).$$

La desigualdad (5.3) implica que:

$$\|I(\rho_{D_A^0}) - I(\rho_{D_B^0})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)W_{1,1}(D_A^0, D_B^0),$$

$$\|I(\rho_{D_A^1}) - I(\rho_{D_B^1})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)W_{1,1}(D_A^1, D_B^1).$$

Finalmente, empleando las desigualdades (5.5) y (5.6) obtenemos que:

$$\|I(\rho_{D_A^0}) - I(\rho_{D_B^0})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)9nm(\|A - B\|_\infty + p) \quad (5.7)$$

$$\|I(\rho_{D_A^1}) - I(\rho_{D_B^1})\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)9nm(\|A - B\|_\infty + p) \quad (5.8)$$

4. Finalmente, se aplanan las matrices  $I(\rho_{D_A^0})$  y  $I(\rho_{D_A^1})$  y se empalman, los dos vectores resultantes obteniendo el vector resultado final  $h_A$ . De manera análoga obtenemos el vector resultado final  $h_B$  a partir de  $I(\rho_{D_B^0})$  y  $I(\rho_{D_B^1})$ . Por las propiedades de la norma infinito de vectores y matrices es claro que:

$$\|h_A - h_B\|_\infty = \max\{\|I(\rho_{D_A^0}) - I(\rho_{D_B^0})\|_\infty, \|I(\rho_{D_A^1}) - I(\rho_{D_B^1})\|_\infty\}.$$

De esta igualdad y de las desigualdades (5.7) y (5.8), concluimos que:

$$\|h_A - h_B\|_\infty \leq \left(5 + \sqrt{\frac{10}{\pi}}60\right)9nm(\|A - B\|_\infty + p)$$

rematando la demostración del teorema. |

Vamos a demostrar un teorema consecuencia del anterior, la estabilidad con paso fijo menor que la precisión de los datos.

**| Teorema 5.6 (Estabilidad con paso menor que la precisión de los datos del método propuesto).** Sean  $A, B \in M_{n,m}(\mathbb{R})$  dos imágenes en escala de grises,  $p > 0$  un paso menor o igual que la precisión de los datos y  $h: C \in M_{n,m}(\mathbb{R}) \mapsto h_C \in \mathbb{R}^k$ , donde  $h_C$  es el resultado de aplicarle a la matriz  $C$  la primera etapa de nuestro método, la descrita en la sección 4.3. Se tiene que:

$$\|h_A - h_B\|_\infty \leq \left( \left( 5 + \sqrt{\frac{10}{\pi}} 60 \right) 18nm \|A - B\|_\infty \right).$$

*Demostración.* Sean  $A, B \in M_{n,m}(\mathbb{R})$  dos imágenes en escala de grises,  $p > 0$  un paso menor o igual que la precisión de los datos y  $h: C \in M_{n,m}(\mathbb{R}) \mapsto h_C \in \mathbb{R}^k$ , donde  $h_C$  es el resultado de aplicarle a la matriz  $C$  la primera etapa de nuestro método.

- Si  $A = B$ ,  $h_A = h_B$ , por tanto, la desigualdad se cumple trivialmente.
- Si  $A \neq B$ , como el paso  $p > 0$  es menor o igual que la precisión de los datos, entonces existe algún elemento de  $A - B$  que es mayor que el paso  $p$  (ya que si no existiese, tendríamos que  $A = B$ ). En consecuencia,  $\|A - B\|_\infty \leq p$ . Como cumplimos las condiciones del teorema 5.5 obtenemos que:

$$\|h_A - h_B\|_\infty \leq \left( \left( 5 + \sqrt{\frac{10}{\pi}} 60 \right) 9nm (\|A - B\|_\infty + p) \right)$$

y como  $(\|A - B\|_\infty + p) \leq 2(\|A - B\|_\infty)$  obtenemos que se cumple la desigualdad a demostrar.

|

### 5.1.2 Carencia de estabilidad del método de Muszynski, G. et al.

Para empezar, vamos a demostrar que la primera etapa del método de Muszynski, G. et al. no es estable cuando el paso tiende a cero. Es decir, vamos a demostrar que, por muy pequeño que sea el paso  $p$ , el algoritmo no va a ser estable. Para demostrarlo, es suficiente con ver que:

$$\|A - B\|_\infty \rightarrow 0, p \rightarrow 0 \not\Rightarrow \|F_A - F_B\|_\infty \rightarrow 0 \quad (5.9)$$

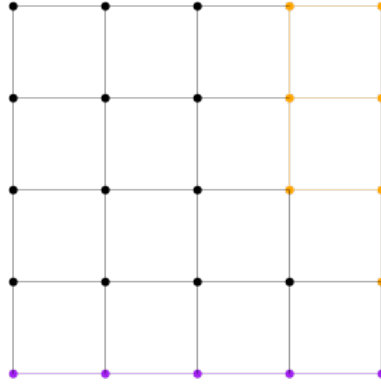


Figura 5.1: Grafo con el que modelamos las rejillas asociadas a las matrices. Las subregiones de interés están coloreadas (la zona naranja representa América y la zona violeta representa las latitudes de Hawái).

donde  $F_A, F_B \in \mathbb{R}^k$  son los vectores obtenidos de aplicarle esta primera etapa, con paso entre umbrales  $p > 0$ , a las matrices  $A, B \in M_{n,m}(\mathbb{R})$ , respectivamente.

Por simplicidad, lo vamos a demostrar con matrices  $5 \times 5$ , pero es claramente extensible a las matrices de datos que estamos usando. Modelamos la rejilla asociada a las matrices como un grafo, considerando las subregiones de interés de la misma forma que hemos hecho en la subsección 3.3.2. Podemos ver este grafo en la figura 5.1.

Consideremos una sucesión de valores  $\delta_n = p_n = \frac{1}{n}$ , con  $n \in \mathbb{N}$ . Para demostrar (5.9), basta con encontrar para cada  $n \in \mathbb{N}$  dos matrices  $A_n, B_n \in M_{5,5}(\mathbb{R})$  tal que  $\|A_n - B_n\|_\infty < \delta_n$  pero que  $\|F_{A_n} - F_{B_n}\|_\infty \geq C$  siendo  $C > 0$  una constante. Remarcar que  $F_{A_n}$  y  $F_{B_n}$  son los vectores obtenidos de aplicarle esta primera etapa, con paso entre umbrales  $p_n > 0$ , a las matrices  $A_n$  y  $B_n$ , respectivamente.

Consideremos, por tanto, las siguientes matrices:

$$A_n = \begin{pmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 25 + \frac{1}{4n} & 25 & 25 \\ 10 & 25 & 25 & 10 & 10 \\ 25 & 25 & 10 & 10 & 10 \end{pmatrix}, \quad B_n = \begin{pmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 25 - \frac{1}{4n} & 25 & 25 \\ 10 & 25 & 25 & 10 & 10 \\ 25 & 25 & 10 & 10 & 10 \end{pmatrix}.$$

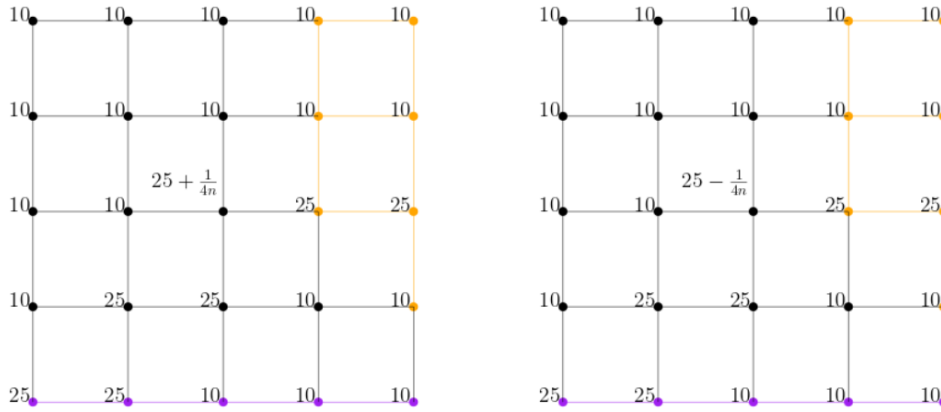


Figura 5.2: Grafos asociados a  $A_n$  (izquierda) y  $B_n$  (derecha). Las subregiones de interés están coloreadas (la zona naranja representa América y la zona violeta representa las latitudes de Hawái).

Es claro que:

$$A_n - B_n = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2n} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

por lo que  $\|A_n - B_n\|_\infty = \frac{1}{2n} < \delta_n$ .

Podemos ver los grafos asociados a  $A_n$  y  $B_n$  en la figura 5.2.

Como el paso entre umbrales es  $p = \frac{1}{n}$ , el conjunto de umbrales considerados,  $U := \{60, 60 - p, 60 - 2p, \dots, 0\}$  (en este orden), es tal que  $25 \in U$  y que  $(25, 25 + \frac{1}{4n}] \cap U = \emptyset$ . Por tanto, resulta claro que el primer subgrafo de nivel superior<sup>1</sup> no vacío aparece, tanto para  $A_n$  como  $B_n$ , con el umbral 25. Este subgrafo de nivel superior con umbral 25 lo podemos ver en la figura 5.3, tanto para  $A_n$  como para  $B_n$ .

Como  $25 \in U$ , va a existir una componente de los vectores  $F_{A_n}$  y  $F_{B_n}$  asociada al subgrafo de nivel superior de umbral 25, denotémosla por  $F_{A_n}^{25}$  y  $F_{B_n}^{25}$  (para  $F_{A_n}$  y

<sup>1</sup>Estamos considerando los subgrafos de nivel superior con  $\geq$ , no con  $>$ .



Figura 5.3: Subgrafos de nivel superior de umbral 25 asociados a  $A_n$  (izquierda) y  $B_n$  (derecha). Las subregiones de interés están coloreadas (la zona naranja representa América y la zona violeta representa las latitudes de Hawái)

$F_{B_n}$ , respectivamente). Si demostramos que  $F_{A_n}^{25} = \frac{7}{25}$  y  $F_{B_n}^{25} = 0$  para todo  $n \in \mathbb{N}$ , concluiríamos que  $\|F_{A_n} - F_{B_n}\|_\infty \geq C = \frac{7}{25} > 0$  y acabaríamos la demostración de inestabilidad.

Veamos que  $F_{A_n}^{25} = \frac{7}{25}$ . El primer umbral para el que el subgrafo de nivel superior tiene alguna componente conexa es 25, y esta componente toca ambas subregiones de interés (las que representan las latitudes de Hawái y el continente americano, coloreadas en figura 5.3), en consecuencia, si que la podemos tener en cuenta. Por lo tanto, como esta componente conexa es la que nace y podemos tener en cuenta antes, el valor de  $F_{A_n}^{25}$  es la proporción de nodos que tiene la componente conexa en el subgrafo de nivel (7) respecto del total de nodos del grafo (25).

Veamos que  $F_{B_n}^{25} = 0$ . Si bien existen dos componentes conexas en el subgrafo de nivel (ver figura 5.3), no podemos tener en cuenta ninguna de esas componentes conexas, ya que ninguna de ellas toca las dos subregiones de interés (las que representan las latitudes de Hawái y el continente americano, coloreadas en la figura). Por tanto, a los ojos del algoritmo, en el umbral 25 aún no apareció ninguna componente conexa a considerar, por lo que  $F_{B_n}^{25} = 0$ .

*Observación 5.1.* Es importante destacar que el método sería inestable aunque se tuviesen en cuenta las componentes conexas que no intersecan ambas subregiones de interés. Si para las mismas sucesiones  $\delta_n$  y  $p_n$  considerásemos:

$$A_n = \begin{pmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 25 + \frac{1}{4n} & 25 + \frac{1}{4n} & 25 + \frac{1}{4n} \\ 10 & 25 + \frac{1}{4n} & 25 + \frac{1}{4n} & 10 & 10 \\ 25 + \frac{1}{4n} & 25 + \frac{1}{4n} & 10 & 10 & 10 \end{pmatrix},$$

$$B_n = \begin{pmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 25 - \frac{1}{4n} & 25 - \frac{1}{4n} & 25 - \frac{1}{4n} \\ 10 & 25 - \frac{1}{4n} & 25 - \frac{1}{4n} & 10 & 10 \\ 25 - \frac{1}{4n} & 25 - \frac{1}{4n} & 10 & 10 & 10 \end{pmatrix},$$

tendríamos que

$$A_n - B_n = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2n} & \frac{1}{2n} & \frac{1}{2n} \\ 0 & \frac{1}{2n} & \frac{1}{2n} & 0 & 0 \\ \frac{1}{2n} & \frac{1}{2n} & 0 & 0 & 0 \end{pmatrix}. \quad (5.10)$$

Obtendríamos, de igual forma, que  $\|A_n - B_n\|_\infty = \frac{1}{2n} < \delta_n$ ,  $F_{A_n}^{25} = \frac{7}{25}$  (ya que el subgrafo de nivel superior de umbral 25 sería el mismo) y que  $F_{B_n}^{25} = 0$  (ya que el subgrafo de nivel superior de umbral 25 sería vacío). Por lo cual, concluimos que también se puede demostrar la inestabilidad sin emplear la regla de que las componentes conexas que no toquen las subregiones de interés no cuentan.

## 5.2 Comparación de resultados numéricos

Para comparar los resultados numéricos del método de Muszynski, G. et al. tenemos que ceñirnos al único conjunto de datos que hemos utilizado ambos: el obtenido del reanálisis MERRA-2 con precisión temporal de  $3h$  y espacial de  $50km$ .

Es importante destacar que mientras los resultados Muszynski, G. et al. se obtuvieron empleando un etiquetado de las imágenes surgido de un método (TECA) distinto al nuestro (TECA-BARD). Estos etiquetados se usan para entrenar la máquina y medir la calidad en la etapa del aprendizaje automático de ambos

métodos, por tanto sesgan los modelos (por tanto, al ser distintos los etiquetados se sesgan de distinta forma). Igualmente, la calidad de las clasificaciones es comparable, aunque con ese matiz.

Queda pendiente, por tanto, realizar la comparación con otros conjuntos de datos y habiendo usado con el mismo etiquetado.

Una vez ya sabemos las condiciones de la comparación podemos confrontar resultados. Comparemos, pues, la calidad de la clasificación del modelo de Muszynski, G. et al. con la de nuestro método, con las distintas posibilidades que hemos considerado anteriormente (utilizando dimensiones de homología 0 y/o 1 y usando KNN o SVM con núcleo radial). En el cuadro 5.1 podemos ver la comparación en términos de exactitud en el conjunto test y en el cuadro 5.2 en términos de sensibilidad y precisión en el conjunto test. En ambos cuadros, como Muszynski, G. et al. rastrean con el *Algoritmo Union-Find* únicamente las componentes conexas (igual que la homología de dimensión 0), consideramos que se usa dimensión 0.

El método propuesto en nuestro trabajo mejora en exactitud en hasta 4 de las posibilidades (en todas las posibilidades en las que se emplea homología de dimensión 0) al método de Muszynski, G. et al.. La posibilidad en la que esa mejora es mejor es la posibilidad propuesta en el trabajo: utilizar homología persistente en ambas dimensiones y KNN. Con esta configuración, se mejora la exactitud en más del 3,2%.

El método propuesto en nuestro trabajo empeora en precisión en todas las posibilidades al método de Muszynski, G. et al., pero, sin embargo, lo mejora en sensibilidad en 5 de las 6 posibilidades. En particular, comparando solo con la posibilidad que hemos propuesto, nuestro método mejora en sensibilidad más de un 15% y empeora en precisión más de un 5,8% respecto del método de Muszynski, G. et al..

Resumidamente, el método propuesto (empleando KNN sobre imágenes de persistencia relativas a dimensiones 0 y 1) obtiene mejores resultados en exactitud (porcentaje de aciertos) y sensibilidad (porcentaje de aciertos en ríos atmosféricos). Sin embargo, obtiene peor resultado en precisión (porcentaje de ríos atmosféricos entre el total de imágenes clasificadas como río atmosférico). Por lo tanto, salvo que se tenga como objetivo principal mejorar la precisión, nuestro método obtiene mejor clasificación que el de Muszynski, G. et al..



Cuadro 5.1: Tabla con las exactitudes (en el conjunto test) obtenidas por el método de Muszynski, G. et al. (el que utiliza en la primera etapa el *Algoritmo Union-Find*) y por nuestro método (empleamos homología persistente en la primera etapa) con las diversas posibilidades en las dimensiones consideradas en homología y el método en la segunda etapa. El método de Muszynski, G. et al. y el que proponemos nosotros (con KNN y usando ambas dimensiones en homología persistente) están destacados en negrita.

Método 1 <sup>a</sup> etapa	Método 2 <sup>a</sup> etapa	Dim. usadas	Exactitud
<b>Alg. Union-Find</b>	<b>SVM Radial</b>	<b>0</b>	<b>0.80</b>
<b>Homología pers.</b>	<b>KNN</b>	<b>0,1</b>	<b>0.8324</b>
Homología pers.	KNN	0	0.8091
Homología pers.	KNN	1	0.7717
Homología pers.	SVM Radial	0,1	0.8310
Homología pers.	SVM Radial	0	0.8162
Homología pers.	SVM Radial	1	0.7183

Cuadro 5.2: Tabla con las distintas precisiones y sensibilidades obtenidas (en el conjunto test) por el método de Muszynski, G. et al. (el que emplea en la primera etapa el *Algoritmo Union-Find*) y por nuestro método (empleamos homología persistente en la primera etapa) con las diversas posibilidades en las dimensiones consideradas en homología y el método en la segunda etapa. El método de Muszynski, G. et al. y el que proponemos nosotros (con KNN y empleando ambas dimensiones en homología persistente) están destacados en negrita.

Método 1 <sup>a</sup> etapa	Método 2 <sup>a</sup> etapa	Dim. usadas	Precisión	Sensibilidad
<b>Alg. Union-Find</b>	<b>SVM Radial</b>	<b>0</b>	<b>0.84</b>	<b>0.74</b>
<b>Hom. pers.</b>	<b>KNN</b>	<b>0,1</b>	<b>0.7816</b>	<b>0.9022</b>
Hom. pers.	KNN	0	0.7643	0.8693
Hom. pers.	KNN	1	0.7109	0.8811
Hom. pers.	SVM radial	0,1	0.8183	0.8320
Hom. pers.	SVM radial	0	0.7970	0.8264
Hom. pers.	SVM radial	1	0.6956	0.7334



## 6 | Conclusiones y trabajo futuro

En este capítulo, en la sección 6.1, explicaremos las conclusiones de nuestro trabajo y, en la sección 6.2, explicaremos el trabajo que queda pendiente.

### 6.1 Conclusiones

Hemos estudiado la importancia del estudio de los ríos atmosféricos (RA) y de su detección. Por ello, hemos analizado el problema de detección de RA a partir de imágenes en escala de grises a lo largo de una región de la superficie de la tierra. Hemos visto que la mayoría de métodos de detección de ríos atmosféricos se basan en fijar umbrales subjetivos de manera previa, sin que haya un consenso científico sobre los umbrales a fijar para dar una definición cuantitativa de lo que es un RA.

Después de ello hemos analizado el método de Muszynski, G. et al., explicado en [7], que combina el análisis topológico de datos (ATD) y el aprendizaje automático para detectar RAs sin tener que fijar umbrales subjetivos en ninguna variable de manera previa. Este método obtiene primero los patrones topológicos de las imágenes y, luego, en base a esos patrones de los datos, clasifica las imágenes según haya o no un río atmosférico en ellas empleando un método de aprendizaje automático, la Máquina Vector Soporte (MVS). La información topológica extraída es la evolución de las componentes conexas subconjuntos de nivel superior. Finalmente, también hemos analizado los resultados de este método.

Seguidamente, hemos construido y propuesto un método de detección que mejora al que acabamos de mencionar. Este método sigue el mismo esquema, pero en la parte del ATD no solo extrae información topológica de la evolución de las componentes conexas de los subconjuntos de nivel superior, sino también de

los huecos. Para ello, modela las imágenes en escala de grises como complejos cúbicos filtrados y genera dos diagramas de persistencia a partir de cada imagen, uno resultado de aplicar homología persistente en dimensión 0 (información sobre evolución de componentes conexas) y el otro de hacerlo en dimensión 1 (información sobre evolución de huecos). Después, cada diagrama se transforma en una imagen de persistencia, con el fin de obtener vectores para poder clasificarlos posteriormente con un método de aprendizaje automático. Posteriormente, con el fin de ver qué método de clasificación supervisada de estos vectores funciona mejor, hemos hecho un procedimiento de Validación Cruzada Anidada *Leave-two-out* y hemos concluido que el método que mejor funciona es el de los  $K$  Vecinos más Cercanos (KNN). Por lo tanto, hemos clasificado estos vectores en *RA* o no *RA* con KNN y hemos medido y analizado la calidad de la clasificación. Finalmente, hemos comprobado que emplear homología persistente en ambas dimensiones obtiene mejores resultados que si solo se utilizase homología persistente en una de las dos dimensiones.

Finalmente, hemos comparado los dos métodos, el analizado y el propuesto, en términos de estabilidad y resultados numéricos. Hemos demostrado que el método que hemos propuesto en este trabajo es estable considerando un paso suficientemente pequeño y que el método ya existente no. Además, al comparar los resultados numéricos de los dos métodos, hemos concluido que nuestro método es el mejor en la mayoría de medidas de bondad de clasificación calculadas. Por tanto, el método que hemos propuesto en este trabajo, es mejor en términos de estabilidad y resultados numéricos, que el método ya existente.

## 6.2 Trabajo futuro

Como trabajo futuro queda hacer un análisis de cuáles son las causas de que falle el algoritmo para ciertas imágenes. Es decir, si hay alguna forma de río atmosférico que no se detecte como *RA*, o viceversa.

Además, queda pendiente, probar nuestro método en más conjuntos de datos, especialmente en los provenientes del modelo CAM5.1. Además, también queda como trabajo futuro probar nuestro método con distintas etiquetas para los datos y ver como se comporta, en particular, con etiquetas hechas manualmente por expertos (con ellas no sesgaríamos nuestro método a otros métodos automáticos) y con las etiquetas surgidas de TECA (para poder hacer una mejor comparación

con el método de Muszynski, G. et al.).

También queda para trabajo futuro, probar nuestro método con distintos pasos en el algoritmo de función de filtrado y ver como evoluciona la calidad de clasificación según su valor. Además, sería interesante, probar el método sin usar el algoritmo de función de filtrado, ya que sin usarlo podríamos obtener más información sobre posibles huecos que tengan los ríos atmosféricos en valores altos de DSVÁ.



# Bibliografía

- [1] Zhu, Y. and Newell, R. E.: *A Proposed Algorithm for Moisture Fluxes From Atmospheric Rivers*. Monthly Weather Review 126:725-735, 1998.
- [2] Ralph, F. M., Dettinger, M.D., , Das, T., Neiman, P.J., and Cayan, D.: *Atmospheric Rivers, Floods, and the Water Resources of California*. Water 3, 455-478, 2011
- [3] Lavers, D. A. and Villarini, G.: The nexus between atmospheric rivers and extreme precipitation across Europe, Geophys. Res. Lett., 40, 3259-3264, 2013.
- [4] Ralph, F. and Dettinger, M.: *Storms, floods, and the science of atmospheric rivers*, Eos T. Am. Geophys. Un., 92, 265, 2011
- [5] Guan, B., Molotch, N. P., Waliser, D. E., Fetzer, E. J., and Neiman, P. J.: *Extreme snowfall events linked to atmospheric rivers and surface air temperature via satellite measurements*, Geophys. Res. Lett., 37, L20401, <https://doi.org/10.1029/2010GL044696>, 2010
- [6] Dettinger, M. D.: Atmospheric rivers as drought busters on the US West Coast, J. Hydrometeorol., 14, 1721-1732, 2013.
- [7] Muszynski, G., Kashinath, K., Kurlin V., Wehner, M., and Prabhat, M.: *Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets*, <https://gmd.copernicus.org/articles/12/613/2019/>, 2019
- [8] Shields, C. A., Rutz, J. J., Leung, L.-Y., Ralph, F. M., Wehner, M., Kawzenuk, B., Lora, J. M., McClenny, E., Osborne, T., Payne, A. E., Ullrich, P., Gershunov, A., Goldenson, N., Guan, B., Qian, Y., Ramos, A.

- M., Sarangi, C., Sellars, S., Gorodetskaya, I., Kashinath, K., Kurlin, V., Mahoney, K., Muszynski, G., Pierce, R., Subramanian, A. C., Tome, R., Waliser, D., Walton, D., Wick, G., Wilson, A., Lavers, D., Prabhat, M., Collow, A., Krishnan, H., Magnusdottir, G., and Nguyen, P.: Atmospheric River Tracking Method Intercomparison Project (ARTMIP): project goals and experimental design, *Geosci. Model Dev.*, 11, 2455-2474, <https://doi.org/10.5194/gmd-11-2455-2018>, 2018
- [9] Liu, Y., Racah, E., Prabhat, M., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M., and Collins, W. D.: *Application of deep convolutional neural networks for detecting extreme weather in climate datasets*, arXiv:1605.01156, 2016
- [10] Tymochko, S., Munch, E., Dunion, J., Corbosiero, K. and Torn, R.: *Using Persistent Homology to Quantify a Diurnal Cycle in Hurricane Felix Pattern Recognition Letters*, 133, 137-143, 2020
- [11] Choe, S. and Ramanna, S.: *Cubical Homology-Based Machine Learning: An Application in Image Classification.*, *Axioms*, 11, 112. <https://doi.org/10.3390/axioms11030112>, 2022
- [12] Kaczynski, T., Mischaikow, K. and Mrozek, M.: *Computational Homology*, volume 157 of Applied Mathematical Sciences. Springer-Verlag, 2004
- [13] Bleile, B., Garin, A., Heiss, T., Maggs, K. and Robins, V.: *The Persistent Homology of Dual Digital Image Constructions.*, arXiv:2102.11397, <https://doi.org/10.48550/arXiv.2102.11397>, 2022
- [14] Edelsbrunner H. and Harer, J.: *Computational Topology, An Introduction*. American Mathematical Society, 2010.
- [15] Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F. and Ziegelmeier, L.: *Persistence Images: A Stable Vector Representation of Persistent Homology*, *Journal of Machine Learning Research*, 18, 16-337, <http://jmlr.org/papers/v18/16-337.html>, 2017
- [16] Russell, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach*. Pearson, London, 2021.



- [17] Dinh, T. L. A. and Aires, F.: Nested leave-two-out cross-validation for the optimal crop yield model selection, *Geosci. Model Dev.*, 15, 3519–3535, <https://doi.org/10.5194/gmd-15-3519-2022>, 2022
- [18] James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning : with Applications in R*. Springer, New York, 2013.
- [19] Hastie, T., Tibshirani, R., and Friedman, J.: *The elements of statistical learning: Data Mining, Inference and Prediction*. Springer Verlag, Berlín, 2009
- [20] Eaton, B.: *User's Guide to the Community Atmosphere Model CAM-5.1*, NCAR, 2011.
- [21] Gelaro, R., McCarty, W., Suárez, M. J., Todling, R., Molod, A., Takacs, L., Randles, C. A., Darmenov, A., Bosilovich, M. G., Reichle, R., Wargan, K., Coy, L., Cullather, R., Draper, C., Akella, S., Buchard, V., Conaty, A., da Silva, A. M., Gu, W., Kim, G.- K., Koster, R., Lucchesi, R., Merkova, D., Nielsen, J. E., Partyka, G., Pawson, S., Putman, S., Rienecker, M., Schubert, S. D., Sienkiewicz, M., and Zhao, B.: *The modern-era retrospective analysis for research and applications, version 2 (MERRA-2)*, *J. Climate*, 30, 5419–5454, 2017
- [22] Prabhat, M., Byna, S., Vishwanath, V., Dart, E., Wehner, M., and Collins, W. D.: *TECA: Petascale pattern recognition for climate science, in: International Conference on Computer Analysis of Images and Patterns*, Springer, 426-436, 2015
- [23] Muszynski, G., Kashinath, K., Kurlin, V., Wehner, M., and Prabhat, M.: *NERSC Science Gateway*, [https://doi.org/10.25342/GMD\\_2018](https://doi.org/10.25342/GMD_2018), 2018
- [24] Hopcroft, J. E. and Ullman, J. D.: *Set merging algorithms*, *SIAM J. Comput.*, 2, 294-303, 1973
- [25] O'Brien, T. A., Risser, M. D., Loring, B., Elbashandy, A. A., Krishnan, H., Johnson, J., Patricola, C. M., O'Brien, J. P., Mahesh, A., Prabhat, M., Arriaga Ramirez, S., Rhoades, A. M., Charn, A., Inda Díaz, H., and Collins, W. D.: *Detection of atmospheric rivers with inline uncertainty quantification: TECA-BARD v1.0.1*, *Geosci. Model Dev.*, 13, 6131–6148, <https://doi.org/10.5194/gmd-13-6131-2020>, 2020

- [26] Unidata, *Network Common Data Form (NetCDF) 1.5.7* [software]. UCAR/Unidata Program Center, Boulder, EE.UU., <https://doi.org/10.5065/D6H70CW6>, 2021
- [27] Bradski, G.: *The OpenCV Library*. Journal of Software Tools, 120, 122-125, 2000
- [28] Dlotko, P.: *Cubical complex*, GUDHI User and Reference, Manual, GUDHI Editorial Board,, [http://gudhi.gforge.inria.fr/doc/latest/group\\_\\_cubical\\_\\_complex.html](http://gudhi.gforge.inria.fr/doc/latest/group__cubical__complex.html), 2015
- [29] Saul, N. and Tralie, C.: *Scikit-TDA: Analysis for Python*, Zenodo.: <http://doi.org/10.5281/zenodo.2533369>, 2019
- [30] Venables, W.N. and Ripley, B.D.: *Modern Applied Statistics with S*, Cuarta Edición, Springer, New York, ISBN 0-387-95457-0 <https://www.stats.ox.ac.uk/pub/MASS4/>, 2002
- [31] Skraba, P. and Turner, K.: *Wasserstein Stability for Persistence Diagrams*, arXiv: Algebraic Topology, <https://doi.org/10.48550/arXiv.2006.16824>, 2020