



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

El uso de funciones de agregación para el desarrollo de un algoritmo de reducción de imagen

Autor/es

PABLO AITOR LIZARRAGA GUERRA

Director/es

DANIEL JOSÉ RODRÍGUEZ LUIS, ANGELA BERNARDINI GAGLIANI y AITOR CASTILLO LÓPEZ

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Matemáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2020-21



El uso de funciones de agregación para el desarrollo de un algoritmo de reducción de imagen, de PABLO AITOR LIZARRAGA GUERRA (publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported. Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.



UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Matemáticas

El uso de funciones de agregación para el desarrollo de un
algoritmo de reducción de imagen

Realizado por:

Pablo Aitor Lizarraga Guerra

Tutelado por:

Daniel José Rodríguez Luis (Universidad de La Rioja)

Aitor Castillo López (Universidad Publica de Navarra)

Angela Bernardini (NAITEC)

Logroño, marzo, 2021

Agradecimientos

A **Daniel José Rodríguez**, por su participación en la redacción de la memoria y por su gran ayuda en momentos críticos en temas ajenos a ella. Para mí es más que un profesor y es una persona que admiro muchísimo.

A **Angela Bernardini**, que aunque tuviera la agenda completa siempre sacó momentos para ayudarme, y eso lo valoro muchísimo.

A **Aitor Castillo, Iosu Rodríguez, Javier Fumanal y Mikel Ferrero**, por guiarme en el mundo de la investigación y sobre todo por los pequeños buenos ratos.

A **Oroel Ipas**, por enseñarme lo interesante que puede llegar a ser la IA y sobre todo por preocuparse en ayudarme en lo máximo posible.

A **Humberto Bustince**, por ayudarme a encontrar mi camino después de la Universidad.

A **Zdenko Takáč**, por su gran teoría matemática que ha ayudado a desarrollar este trabajo.

A **NAITEC** y a la **UPNA**, por proporcionarme todos los medios necesarios para el desarrollo de este trabajo y darme la oportunidad de seguir realizando trabajos de investigación.

A **Marta Valdazo**, a mi madre y a todos mis amigos por preguntar y escuchar sobre mi memoria aunque no entendiesen la mitad de las cosas.

Resumen

En el contexto de la imagen digital, el proceso de reducción de imagen consiste en disminuir las dimensiones de una imagen preservando la mayor información de la imagen original. Algunas de las aplicaciones del método de reducción de imagen van desde mejorar los tiempos de espera producidos por los cambios de una imagen (cambios de coloración, cambios de saturación, aplicación de filtros, etc.) hasta reducir el coste de almacenamiento y transmisión de los archivos.

En este trabajo presentamos los fundamentos matemáticos de un algoritmo de reducción de imagen basado en lo que se conoce como funciones de agregación sobre el conjunto $[0, 1]^3$. Este nuevo método para el tratamiento de las imágenes forma parte del proyecto **SARS2DETECT: Desarrollo de una herramienta de diagnóstico evolutivo para el daño inducido por el nuevo coronavirus** (actualmente en desarrollo). Este proyecto está liderado por NAITEC con UPNA y Complejo Hospitalario Navarro en calidad de socios.

Este trabajo es el resultado de una colaboración conjunta entre la Universidad de La Rioja (UR), la Universidad Pública de Navarra (UPNA) y el Centro Tecnológico NAITEC.

Abstract

In the context of digital imaging, the image reduction technique consists of reducing the dimensions of a image preserving as much as possible information from the original image. Some applications of the image reduction method go from accelerate computation on an image (color changes, saturation changes, filters, etc.) to reducing the cost of its storage and transmission.

In this work we present the mathematical foundations of an image reduction algorithm based on the well-known aggregation functions on the set $[0, 1]^3$. This new method is part of the **SARS2DETECT: Development of an evolutionary diagnostic tool for damage induced by the new coronavirus** (currently in development). This project is led by NAITEC with UPNA and Complejo Hospitalario Navarro as partners.

This is a joint work with University of La Rioja (UR), the Public University of Navarra (UPNA) and the NAITEC Technological Center.

Índice general

1. Introducción	9
2. Preliminares	11
2.1. Funciones de agregación	11
2.1.1. Construcción de órdenes admisibles	13
2.2. Imágenes digitales	15
2.2.1. Reducción y magnificación de imagen	17
2.2.2. Métodos de comparación de imágenes	18
2.3. Redes neuronales	20
2.3.1. Aprendizaje automático	20
2.3.2. Introducción a redes neuronales artificiales	20
2.3.3. Redes neuronales convolucionales	21
3. Fundamentos matemáticos del algoritmo de reducción de imagen	23
3.1. Funciones de agregación en L^2	23
3.2. Funciones de agregación en L^3	28
3.3. Construcción de funciones de agregación respecto a un orden admisible	32
4. Algoritmo de reducción de imágenes y experimentación	37
4.1. Aplicación del algoritmo: Proyecto SARS2DETECT	42
5. Conclusiones	45

Capítulo 1

Introducción

La reducción de imagen ha sido un tema muy estudiado durante los últimos años en el ámbito de la imagen digital. Esta consiste en disminuir la resolución de una imagen intentando mantener la máxima información posible de la imagen original. Por ejemplo, mediante un algoritmo basado en splines con un método de diferencias finitas que permite el cálculo de productos internos con funciones de análisis, obtenemos un método de reducción de imagen como puede verse en [3]. La reducción de imagen y su proceso inverso, la magnificación, es un tema muy estudiado debido a su gran número de aplicaciones, entre las que se encuentra la comparación de imágenes capturadas con distintos sensores (microscopios electrónicos, huellas digitales, etc.), las imágenes del satélite (como Google Maps), así como en la visualización de imágenes médicas, como se observa en [8].

Una de las aplicaciones más cotidianas del proceso de reducción de imagen se puede encontrar en los dispositivos móviles, donde las imágenes almacenadas deben cambiar de tamaño para poder ser visualizadas en pantallas generalmente pequeñas de dichos dispositivos. También, el proceso de reducción está presente en el preprocesamiento de imágenes para su uso en algoritmos de aprendizaje automático que pueden resultar costosos tanto computacional como temporalmente.

En la literatura existen muchos métodos de reducción diferentes según el tratamiento que le dan a los píxeles de la imagen. Por un lado, están aquellos que reducen la imagen globalmente, como por ejemplo [7] que utiliza la técnica de transformadas difusas directas e inversas. Por otro lado, existen otros métodos que dividen la imagen en piezas más pequeñas y actúan sobre cada una de ellas localmente, como por ejemplo [9] que usa un método basado en el concepto de la transformada difusa discreta de una función definida en un intervalo real con respecto a los conjuntos difusos. Estos últimos métodos que dividen los píxeles en unidades más pequeñas permite diseñar algoritmos muy eficientes en el tiempo, manteniendo algunas de las propiedades específicas más importantes de la imagen original.

Basándonos en [1], en este trabajo se propone un algoritmo de reducción de imagen por piezas, para llevar a cabo la reducción de imágenes, y con el propósito de mejorar algunos de los resultados obtenidos por otros métodos de reducción de imagen ya existentes.

Además de la reducción de imagen, tanto este algoritmo como aquellos que se puedan deducir de la teoría presente en [1] pueden ser utilizados para temas relacionados con el entrenamiento de redes neuronales convolucionales.

El objetivo de este trabajo es aplicar el algoritmo de reducción a imágenes en el ámbito sanitario, más concretamente a imágenes de radiografías de tórax. La idea es, partiendo de una imagen original de una radiografía de tórax, proporcionar mediante el algoritmo de reducción una imagen de menores dimensiones pero manteniendo las propiedades más importantes del archivo original, como pueden ser

los bordes o la opacidad. Una vez aplicado el algoritmo de reducción a una colección considerable de imágenes, se procederá a introducir dichas reducciones en una red neuronal convolucional, lo que favorecerá a un aprendizaje automático de nuestra red con un menor coste temporal, una mayor eficiencia computacional y con un resultado similar al que se obtiene con las imágenes originales.

La presente memoria está estructurada en 5 capítulos, incluyendo la Introducción. El Capítulo 2 está dedicado a los preliminares, donde hablaremos de los conceptos básicos para poder definir una función de agregación acompañado de algunas propiedades y unos cuantos ejemplos. Así mismo explicaremos brevemente los procesos de reducción y magnificación de imágenes y también introduciremos las redes neuronales, en concreto las redes neuronales convolucionales.

Posteriormente, en el Capítulo 3, desarrollamos la teoría presente en [1] acompañada de algunos ejemplos ilustrativos. Así mismo, veremos el teorema sobre el que se fundamenta nuestro algoritmo de reducción de imagen.

Más tarde, en el Capítulo 4 realizaremos pruebas con este algoritmo y haremos comparaciones con algunos de los métodos de reducción existentes en la literatura. Al término de este capítulo mostraremos los resultados de la reducción de las imágenes radiográficas de pulmón de pacientes con COVID-19.

Finalmente el Capítulo 5 está dedicado a las conclusiones y al planteamiento de futuras líneas de investigación que se derivan de este trabajo.

Capítulo 2

Preliminares

En este capítulo empezamos presentando algunos conceptos básicos que tienen una fundamentación matemática y que vienen acompañados de ejemplos ilustrativos. Estos conceptos serán de gran utilidad para comprender el algoritmo sobre el que se fundamenta el algoritmo de reducción de imágenes y del que hablaremos en el Capítulo 4.

Por último, haremos una breve descripción sobre los fundamentos de los procesos de reducción y magnificación de imágenes, así como una introducción a las redes neuronales y el aprendizaje automático.

2.1. Funciones de agregación

Definición 2.1. Un **orden parcial** es una relación binaria R sobre un conjunto no vacío X que es reflexiva, antisimétrica y transitiva, es decir, para cualquier a, b , y c en X se tiene que:

- aRa (reflexividad)
- Si aRb y bRa , entonces $a = b$ (antisimetría)
- Si aRb y bRc , entonces aRc (transitividad)

Denotando R como \leq , entonces un conjunto X parcialmente ordenado lo denotamos por (X, \leq)

A lo largo de esta memoria, consideraremos (L, \leq) el conjunto parcialmente ordenado donde $L = [0, 1]$ y \leq es el orden natural en \mathbb{R} . Denotamos $\mathbf{0} := (0, \dots, 0) \in L^n$ y $\mathbf{1} := (1, \dots, 1) \in L^n$ y denotamos con letras mayúsculas los elementos en L^n , esto es, $X := (x_1, \dots, x_n) \in L^n$.

A continuación, introducimos la noción de función de agregación que permite, como veremos más adelante, inducir un orden sobre el conjunto L^n .

Definición 2.2. Una **función de agregación n-aria M en L** es una función $f : L^n \rightarrow L$ creciente en cada variable que satisface que $M(\mathbf{0}) = 0$, y que $M(\mathbf{1}) = 1$.

Dentro de las funciones de agregación, nos centramos en aquellas que son idempotentes y simétricas según las siguientes definiciones. De hecho, las propiedades antes mencionadas serán las que se exigirán a las funciones de agregación para desarrollar el algoritmo de reducción propuesto en el Capítulo 4.

Definición 2.3. Una función de agregación M se dice **idempotente** si para todo $X = (t, \dots, t) \in L^n$ se cumple que $M(X) = t$.

Definición 2.4. Una función de agregación M se dice **simétrica** si su valor no depende de la permutación de sus argumentos, esto es si

$$M(x_1, \dots, x_n) = M(x_{P(1)}, \dots, x_{P(n)})$$

para todo $X \in L^n$ y toda permutación $P = (P(1), \dots, P(n))$ de $(1, \dots, n)$

A continuación, veremos algunos ejemplos ilustrativos de funciones de agregación en L^n . Algunos de estas funciones serán de gran utilidad en el desarrollo del algoritmo de reducción de imagen presente en el Capítulo 4.

Ejemplo 2.1. Las funciones máximo y mínimo son funciones de agregación. Sea $X = (x_1, \dots, x_n)$ con $x_i \in L$ se definen:

- $\vee(\vec{X}) =: \text{máx}(X) = \max_{i=1, \dots, n} x_i$
- $\wedge(\vec{v}) =: \text{mín}(X) = \min_{i=1, \dots, n} x_i$

Ejemplo 2.2. Dado un vector de ponderación $\vec{w} = (w_1, \dots, w_n)$ donde $w_i \in L$ y $\sum_{i=1}^n w_i = 1$, la **media aritmética ponderada** es la función

$$M_{\vec{w}}(X) = \sum_{i=1}^n w_i x_i$$

Ejemplo 2.3. La **mediana** es la función

$$\text{Med}(X) = \begin{cases} \frac{1}{2}(x_{(k)} + x_{(k+1)}) & , \text{ si } n = 2k \\ x_{(k)} & , \text{ si } n = 2k - 1 \end{cases}$$

donde $x_{(k)}$ es la componente del vector X_{\searrow} siendo éste, el vector X ordenado de manera no creciente.

Ejemplo 2.4. Dado un vector de ponderación $\vec{w} = (w_1, \dots, w_n)$, la función **OWA** (siglas en inglés de **Ordered Weighted Averaging**) es:

$$OWA_{\vec{w}}(X) = \sum_{i=1}^n w_i x_{(i)}$$

donde $x_{(i)}$ son las coordenadas del vector X_{\searrow} y donde el vector de ponderación que viene dado por:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right)$$

siendo:

$$Q(r) = \begin{cases} 0 & , \text{ si } r < a \\ \frac{r-a}{b-a} & , \text{ si } a \leq r \leq b \\ 1 & , \text{ si } r > b \end{cases}$$

con $a, b \leq 1$ fijos.

Definición 2.5. Sea $\mathcal{N} = \{1, 2, \dots, n\}$. Una **medida difusa discreta** es una función $v : 2^{\mathcal{N}} \rightarrow [0, 1]$ que es monótona ($v(\mathcal{A}) \leq v(\mathcal{B})$ cuando $\mathcal{A} \subset \mathcal{B}$) y satisface $v(\emptyset) = 0$ y $v(\mathcal{N}) = 1$

Ejemplo 2.5. La **integral Choquet discreta** con respecto a una medida difusa v es la dada por

$$C_v(X) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)})v(A_i)$$

donde $A_i \subset X$, con $i = 1, 2, \dots, n$, $X_{\nearrow} = (x_{(1)}, x_{(2)}, \dots, x_{(n)})$ es una permutación no decreciente de X , y $x_{(0)} = 0$.

Ejemplo 2.6. La **integral Sugeno discreta** con respecto a una medida difusa v es la dada por

$$S_v(X) = \bigvee_i \wedge (x_{(i)} v(A_i))$$

donde $A_i \subset X$, con $i = 1, 2, \dots, n$, $X_{\nearrow} = (x_{(1)}, x_{(2)}, \dots, x_{(n)})$ es una permutación no creciente de X , y $x_{(0)} = 0$.

Es fácil comprobar que todos los ejemplos descritos cumplen las propiedades de idempotencia y simetría.

2.1.1. Construcción de órdenes admisibles

En esta sección veremos cómo utilizar las funciones de agregación para definir nuevos órdenes sobre el conjunto L^n .

En este punto, recordamos que un orden total sobre un conjunto no vacío X es un orden parcial (según la Definición 2.1) con la propiedad de completitud.

Definición 2.6. Un **orden total** es una relación binaria R sobre un conjunto no vacío X que es reflexiva, antisimétrica, transitiva y completa, es decir, para cualquier a, b , y c en X se tiene que:

- aRa (reflexividad)
- Si aRb y bRa , entonces $a = b$ (antisimetría)
- Si aRb y bRc , entonces aRc (transitividad)
- $a \leq b$ ó $a \geq b$

Consideramos el orden parcial \leq_P inducido por \leq dado de la siguiente manera: $X \leq_P Y$ si y solo si $x_i \leq y_i$ para todo $\{1, \dots, n\}$

En general, sobre el conjunto L^n es posible definir distintos órdenes totales. Ahora bien, cuando un orden total es compatible, en cierto sentido, con el orden natural parcial entonces se dice que el orden es admisible según la siguiente definición.

Definición 2.7. Sea n un entero positivo. Un orden total \leq_L en L^n se dice **admisible** si, para todo $X = (x_1, \dots, x_n), Y = (y_1, \dots, y_n) \in L^n$, tales que $x_i \leq y_i$ para todo $i \in \{1, \dots, n\}$ se tiene que $X \leq_L Y$

Basándonos en [1], estamos interesados en órdenes admisibles que vengan definidos a partir de funciones de agregación en L^n . Por ello, veamos cómo una función de agregación induce un orden en el conjunto L^n el cual, bajo ciertas condiciones, podremos asegurar que es admisible.

Definición 2.8. Sea n un entero positivo y $\mathbf{M} = (M_1, \dots, M_n)$ una secuencia de n n-arias funciones de agregación $M_i : L^n \rightarrow L$. Dados $X, Y \in L^n$, definimos el **orden \leq_M inducido por \mathbf{M}** como:

- $X <_M Y$ si y solo si existe $k \in \{1, \dots, n\}$ tal que $M_j(X) = M_j(Y)$ para todo $j \in \{1, \dots, k-1\}$ y $M_k(X) < M_k(Y)$.

- $X \leq_M Y$ si y solo si $X <_M Y$ o $X = Y$

A continuación, veamos algunos ejemplos de órdenes inducidos por funciones de agregación.

Ejemplo 2.7. En el conjunto L^3 sean M_1, M_2, M_3 la media aritmética, la mediana y máximo, respectivamente.

Observamos que:

1. Si $\mathbf{M} = (M_1, M_2)$, para $X = (0.1, 0.2, 0.3)$ e $Y = (0.4, 0.5, 0.6)$ se tiene que $M_1(X) = 0.2$ mientras que $M_1(Y) = 0.5$. Por lo tanto ya que $M_1(X) < M_1(Y)$ podemos afirmar que $X <_{\mathbf{M}} Y$. Además, es fácil comprobar que el orden inducido por \mathbf{M} es admisible.
2. Si $\mathbf{M} = (M_2, M_3)$ para $X = (0.1, 0.2, 0.2)$ e $Y = (0.2, 0.2, 0.2)$ se tiene que $M_2(X) = M_2(Y) = 0.2$, $M_3(X) = M_3(Y) = 0.2$. En este caso el orden $\leq_{\mathbf{M}}$ inducido por \mathbf{M} no es admisible ya que no se cumple que $X <_{\mathbf{M}} Y$, ni tampoco $X = Y$

El siguiente resultado permite caracterizar aquellos órdenes inducidos por funciones de agregación que son admisibles en el conjunto L^n siempre y cuando cada una de las componentes de la función de agregación $\mathbf{M} = (M_1, \dots, M_n)$ sea inyectiva en el conjunto L^n .

Proposición 2.1. Sea n un entero positivo y $\mathbf{M} = (M_1, \dots, M_n)$ una secuencia de n funciones de agregación $M_i : L^n \rightarrow L$. El orden \leq_M inducido por \mathbf{M} es un orden admisible en L^n si y solo si las funciones de agregación satisfacen que $X = Y$ siempre y que $M_i(X) = M_i(Y)$, $\forall i \in \{1, \dots, n\}$

Demostración. Trivial siguiendo la Definición 2.7 y usando la Definición 2.8 □

En este trabajo, todas las funciones $M_i(X)$, para todo $i \in \{1, \dots, n\}$, serán medias aritméticas ponderadas con vectores de ponderación normalizados:

$$M_i(X) = \alpha_{i1}x_1 + \dots + \alpha_{in}x_n \quad \forall i \in \{1, \dots, n\} \quad (2.1)$$

donde $\alpha_{i1} + \dots + \alpha_{in} = 1 \quad \forall i \in \{1, \dots, n\}$.

De hecho, el siguiente resultado permite caracterizar las funciones de agregación cuyas componentes sean medias aritméticas ponderadas en términos de la regularidad de la matriz donde las filas son, precisamente, las componentes de los vectores de ponderación.

Proposición 2.2. Sea $\mathbf{M} = (M_1, \dots, M_n)$ una secuencia de n funciones de agregación dadas como en la ecuación (2.1). El orden $\leq_{\mathbf{M}}$ es un orden admisible en L^n si y solo si la matriz

$$A_{\mathbf{M}} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix}$$

es regular. En este caso decimos que $\leq_{\mathbf{M}}$ es un orden inducido por la matriz $A_{\mathbf{M}}$ y escribimos $\leq_{A_{\mathbf{M}}}$

Demostración. Dadas M_i funciones de agregación como en la ecuación (2.1), consideramos la aplicación lineal

$$T_{\mathbf{M}}(X) = (M_1(X), \dots, M_n(X)), X \in L^n$$

Es claro que $T_{\mathbf{M}}$ es inyectiva si y solo si cada M_i también es inyectiva. Observamos que la regularidad de la matriz $A_{\mathbf{M}}$ asociada a la aplicación $T_{\mathbf{M}}$ es equivalente al hecho de que $T_{\mathbf{M}}$ sea inyectiva. Luego, el resultado se sigue como aplicación de la Proposición 2.1. □

A continuación, veamos algunos ejemplos del resultado anterior

Ejemplo 2.8. En el conjunto L^3 , sea $\leq_{\mathbf{M}}$ el orden inducido por la función de agregación $\mathbf{M} = (M_1, M_2, M_3)$ donde

- $M_1(x_1, x_2, x_3) = \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3$
- $M_2(x_1, x_2, x_3) = \frac{1}{6}x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3$
- $M_3(x_1, x_2, x_3) = \frac{1}{5}x_1 + \frac{1}{5}x_2 + \frac{3}{5}x_3$

Observamos que la matriz A_M generada por los anteriores coeficientes es regular al ser $\det(A_M) = 2/45$. Por lo tanto el orden \leq_M inducido por la función de agregación M es admisible.

Ejemplo 2.9. En el conjunto L^3 , consideremos el orden $\leq_{\mathbf{M}}$ inducido por la función de agregación $\mathbf{M} = (M_1, M_2, M_3)$, siendo

- $M_1(x_1, x_2, x_3) = \frac{1}{6}x_1 + \frac{1}{6}x_2 + \frac{2}{3}x_3$
- $M_2(x_1, x_2, x_3) = \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3$
- $M_3(x_1, x_2, x_3) = \frac{1}{8}x_1 + \frac{1}{8}x_2 + \frac{3}{4}x_3$

Es fácil observar que $M_3(x_1, x_2, x_3)$ es una combinación lineal de $M_1(x_1, x_2, x_3)$ y $M_2(x_1, x_2, x_3)$, por lo que la matriz A_M generada por los anteriores coeficiente no es regular. Por lo tanto, el orden \leq_M definido en este ejemplo no es admisible.

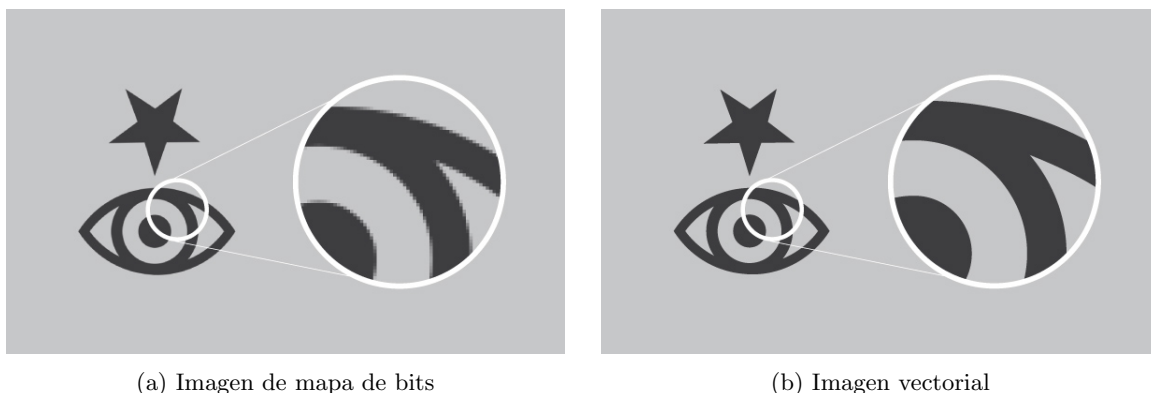
Los anteriores resultados, aunque en otro contexto, pueden encontrarse en [10].

2.2. Imágenes digitales

Una **imagen digital** es una representación bidimensional de una imagen a partir de una matriz numérica. Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de bits) o de un gráfico vectorial.

Las **imágenes de mapa de bits**, están formadas por una cuadrícula de puntos llamados píxeles, que se organizan en una rejilla. Cada uno de esos píxeles tiene un color definido por un valor. Si ampliamos una imagen de mapa de bits veremos cada uno de los píxeles con sus colores, tal y como ocurre en la Figura 2.1 (a). Cuantos más píxeles tenga una imagen en mapa de bits mayor será el tamaño de la cuadrícula y mejor será la calidad de la imagen. En este caso las imágenes tendrán un tamaño en píxeles de $N \times M$ píxeles, teniendo cada píxel p componentes que dependen del modelo de color que elijamos como veremos mas adelante.

Las **imágenes vectoriales** se basan en coordenadas matemáticas que definen su posición, su forma, su color y otros atributos. Están formadas por vectores, que son elementos geométricos como puntos, líneas, polígonos o segmentos. Si ampliamos una imagen vectorial veremos cada una de los trazos y de los colores sin pérdida de calidad, tal y como ocurre en la Figura 2.1 (b).



(a) Imagen de mapa de bits

(b) Imagen vectorial

Figura 2.1: Tipos de imágenes digitales

Las imágenes de mapa de bits representan fielmente la realidad con un alto nivel de detalle pero a la hora de reducir o aumentarlas pueden perder demasiada calidad. En cambio las imágenes vectoriales pueden escalarse sin perder nada de calidad, aunque tienen grandes limitaciones para crear imágenes reales, ya que para ello los archivos que llegan a representar una imagen casi de forma perfecta son extremadamente complejos y pesados.

El mapa de bits es el formato más utilizado para el tratamiento de imagen y el que usaremos nosotros.

Para representar las imágenes en mapa de bits usaremos lo que se conoce como un **modelo de color**, esto es, un sistema estructurado para crear una gama completa de colores de un pequeño conjunto de colores primarios definidos. Hay tres modelos fundamentales de la teoría del color:

- el modelo de luces de color rojo, verde y azul **RGB**(siglas en inglés de **R**ed, **G**reen, **B**lue), originalmente explorado por Isaac Newton en 1666;
- el modelo de colores cian, magenta, amarillo y negro clave **CMYK**(siglas en inglés de **C**yan, **M**agenta, **Y**ellow y **K**ey (negro)) patentado originalmente por Jacob Christoph Le Blon en 1719
- el modelo de colores rojo, amarillo y azul completamente resumido por Johann Wolfgang von Goethe en 1810.

El modelo de color RGB ensambla las luces primarias de rojo, verde y azul juntas en varias combinaciones para producir un amplia gama de colores. Dicho modelo se denomina modelo de color aditivo en el que la combinación de rojo, verde y azul producen luz blanca y que se encuentra presente en diversas tecnologías que producen imágenes en color, como la fotografía convencional y la visualización de imágenes en sistemas electrónicos. Ejemplos en donde se usa este modelo son:

- Dispositivos de entrada: escáneres de imágenes, videojuegos y cámaras digitales, así como cámaras de televisión y vídeo.
- Dispositivos de salida: televisores, pantallas de ordenador y teléfono móvil

El modelo de color RGB tiene un lógica sólida en términos de la física y la percepción humana de los colores. En este modelo, para indicar con qué proporción es mezclado cada color, se asigna un valor a cada uno de los colores primarios, de manera que el valor 0 significa que no interviene en la mezcla y, a medida que ese valor aumenta, aporta más intensidad a la mezcla. El intervalo de valores se codifica con un byte (8 bits). Así, la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255 y cada color es definido por un conjunto de valores escritos entre paréntesis

(correspondientes a valores **R**, **G** y **B**) y separados por comas como vemos en la Figura 2.2 (a). Además en la Figura 2.2 (b) observamos como las diferentes capas de una imagen forman la imagen. En este trabajo usaremos el modelo RGB cuyas componentes, una vez normalizadas, serán los datos sobre los que aplicar el algoritmo de reducción de imágenes.

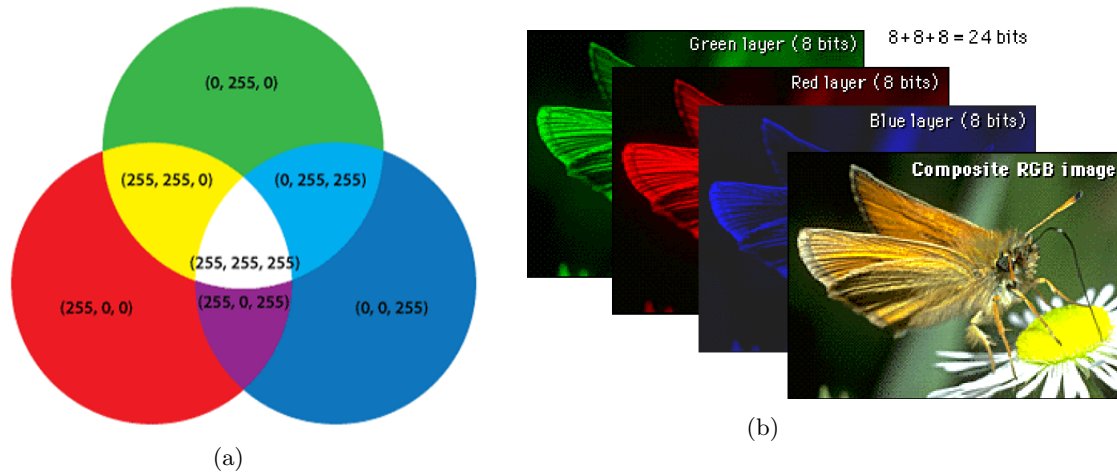


Figura 2.2: Representación y capas del modelo RGB

En la siguiente sección, hablaremos brevemente del proceso de reducción y magnificación de imágenes.

2.2.1. Reducción y magnificación de imagen

La **reducción de imagen** consiste en disminuir la resolución ("tamaño") de una imagen intentando mantener la máxima información posible de la imagen original.

Existen dos maneras de reducir una imagen. La primera manera es globalmente y es recomendable aplicarla solo a imágenes vectoriales, ya que son imágenes compuestas por formas y únicamente habría que reducir cada una de ellas. La segunda es reducir la imagen de manera local, que consiste en dividir la imagen en ventanas de píxeles de diferentes tamaños y aplicar la reducción a cada una de ellas. Esta última forma es la que se utiliza para las imágenes en mapa de bits y la que nosotros usaremos en este trabajo. Para perder la menor información posible sobre la imagen en la reducción es necesario desarrollar un método eficiente, y es por ello que dentro del tratamiento de imagen es un tema muy estudiado.

Como vemos en la Figura 2.3, para la reducción local la imagen se divide en piezas, que son ventanas de $n \times m$ píxeles, y a cada una de ellas se les aplica una función de agregación que reduce esa ventana a un único píxel. Se hace esto por cada una de las profundidades de cada píxel.

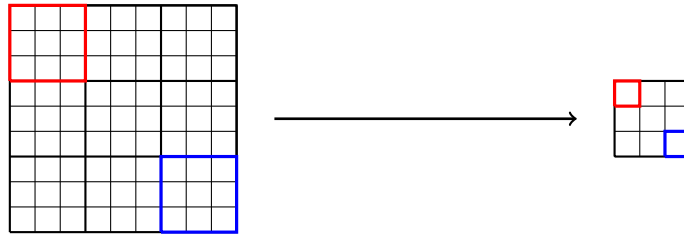


Figura 2.3: Ejemplo de reducción de una matriz de píxeles

Como hemos señalado anteriormente, las funciones de agregación en el conjunto L^n aparecen en muchos métodos de reducción de imágenes (véase [5]). Entre las funciones de agregación más comunes se encuentran: máximo, mínimo, media aritmética, mediana, OWA, integral Choquet e integral Sugeno.

La **magnificación de imagen** es el proceso inverso a la reducción. Para magnificar una imagen de tamaño $N \times M$ a $(n \times N) \times (m \times M)$ creamos $N \times M$ ventanas de tamaño $n \times m$ para obtener ventanas vacías del tamaño de magnificación deseado. Éstas ventanas se pueden rellenar de dos maneras:

- colocando cada píxel i, j en su ventana correspondiente $n \times m$ veces
- rellenando cada ventana con la información de cada píxel correspondiente y sus píxeles mas cercanos como puede verse en la Figura 2.4.

En este trabajo magnificaremos de la segunda manera utilizando el algoritmo desarrollado en [2].

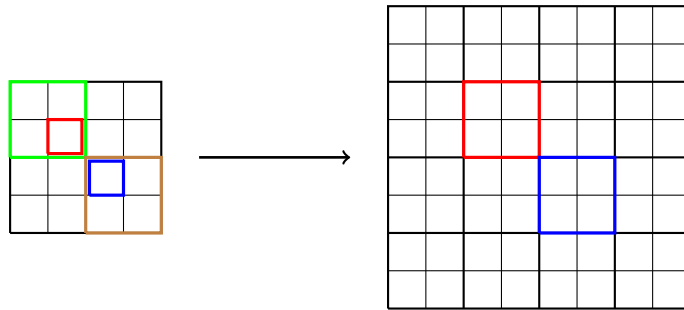


Figura 2.4: Ejemplo de magnificación de una matriz de píxeles

2.2.2. Métodos de comparación de imágenes

Para ver la calidad de una imagen reducida, se compara la imagen original con la imagen magnificada a tamaño original de la imagen reducida. Los métodos de comparación que usaremos en este trabajo son:

1. La relación pico de señal a ruido o **PSNR** (siglas en inglés de **Peak Signal-to-Noise Ratio**): mide la calidad de reconstrucción (magnificación) de la imagen reducida. La señal son los datos originales y el ruido es el error introducido por la compresión. Al comparar compresiones de imágenes, el PSNR es una aproximación a la percepción humana de la calidad de la reconstrucción. Sus valores van desde 0 hasta 100

Dada una imagen $m \times n \times 3$ libre de ruido I y su aproximación ruidosa K , el error cuadrático medio (MSE) se define como:

$$MSE = \frac{1}{3mn} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \sum_{k=0}^2 [I(i, j, k) - K(i, j, k)]^2$$

Y el PSNR (en dB) se define como:

$$PSNR(I, J) = 20 \log_{10}(MAX_I) - 10 \log_{10}(MSE)$$

donde MAX_I es el valor máximo posible de un píxel de la imagen (en nuestro caso 255 ó normalizado 1).

Una reducción es buena si el valor del PSNR está en torno a 30 db (± 5)

2. El índice de similitud estructural o **SSIM**(Structural Similarity Index Measure) entre la imagen reconstruida J y la imagen original I. Se define como:

$$SSIM(I, J) = \frac{(2\mu_I\mu_J + c_1)(2\sigma_{IJ} + c_2)}{(\mu_I^2 + \mu_J^2 + c_1)(\sigma_I^2 + \sigma_J^2 + c_2)}$$

donde

- μ_I es la media de los valores de I
- μ_J es la media de los valores de J
- σ_I^2 es la varianza de los valores de I
- σ_J^2 es la varianza de los valores de J
- σ_{IJ} es la covarianza de los valores de I y J
- $c_1 = (0.01S)^2$; $c_2 = (0.03S)^2$ son dos variables para estabilizar la división con denominador débil
- S es el valor máximo posible de un píxel de la imagen (en nuestro caso 255 ó normalizado 1).

Una reducción es buena si este índice es mayor que 0.90.

3. **El índice de similitud entre imágenes o ISI** (Index of Similarity between Images): de una imagen de tamaño $m \times n \times 3$ libre de ruido I y su aproximación ruidosa K, se define como:

$$ISI = 1 - \frac{1}{3mn} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \sum_{k=0}^2 |I(i, j, k) - K(i, j, k)|$$

Una reducción es buena si este índice es mayor que 0.90.

Decidir si una reducción es buena o no dependiendo de los valores obtenidos aplicando los anteriores métodos de comparación no suele ser muy concluyente ya que no es lo mismo reducir una imagen a un tamaño u otro. Tendremos en cuenta esos valores, pero como nosotros compararemos diferentes métodos de reducción, nos fijaremos más en la diferencia de resultados entre los diferentes métodos que apliquemos.

2.3. Redes neuronales

Como hemos comentado en la Introducción de esta memoria, uno de los pasos de este proyecto es utilizar una red neuronal para clasificar unas radiografías de pulmón. Por ello explicaremos brevemente que es una red neuronal.

2.3.1. Aprendizaje automático

Antes de entrar en detalle a explicar cómo funcionan las redes neuronales haremos una breve introducción al aprendizaje automático. Podemos definir el **aprendizaje automático (Machine Learning)** como una rama de la inteligencia artificial que proporciona a las computadoras la capacidad de aprender sin ser programadas explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden cambiar cuando se exponen a nuevos datos.

El aprendizaje automático se clasifica principalmente en dos tipos:

- **Aprendizaje supervisado:** Deducción de una función a partir de pares de objetos, de los cuales una componente del par son los datos de entrada y el otro los resultados deseados, para conseguir clasificar también nuevos datos que pudiera recibir.
- **Aprendizaje no supervisado:** Trata los objetos de entrada como un conjunto de variables aleatorias, intentando aprender características o patrones que permita clasificar los datos.

Nosotros nos centraremos en las redes neuronales artificiales que son un tipo de aprendizaje supervisado como veremos en la siguiente sección.

2.3.2. Introducción a redes neuronales artificiales

Una **red neuronal artificial** es una técnica de aprendizaje automático supervisado, empleada en la resolución de muchos de los problemas de la actualidad, que emula el funcionamiento de una red neuronal biológica, para que a partir de unos datos de entrada obtengamos unos resultados de salida.

La unidad básica es la neurona y la unión de varias de ellas creará la red. La información de entrada atraviesa la red neuronal, donde se somete a diversas operaciones, produciendo unos valores de salida tal y como vemos en la Figura 2.5.

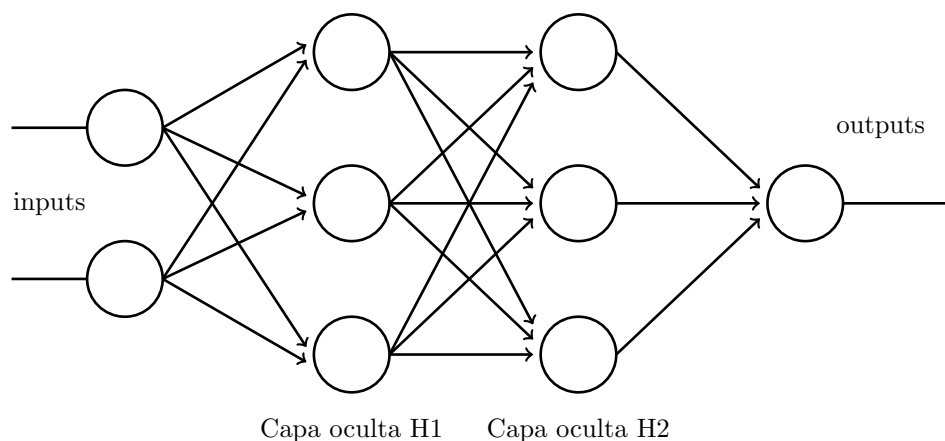


Figura 2.5: Estructura de una red neuronal

Para que la salida de la red sea lo mas acertada posible, deberá entrenarse primero. En este trabajo no nos centramos en el entrenamiento de una red neuronal, pero para mas información sobre ello visitar [13].

2.3.3. Redes neuronales convolucionales

Las **redes neuronales convolucionales** (CNNs) son un tipo de red Neuronal artificial con aprendizaje supervisado que procesa sus capas imitando al córtex visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y “ver”. Para ello, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.

Lo que distingue a las redes neuronales convolucionales son las llamadas **convoluciones**: Estas consisten en tomar grupos de píxeles cercanos de la imagen de entrada e ir operando matemáticamente (producto escalar) contra una pequeña matriz que se llama kernel. Suponiendo que ese kernel tiene un tamaño de $n \times n$ píxeles, logra visualizar todas las neuronas de entrada (de izquierda-derecha, de arriba-abajo) y así logra generar una nueva matriz de salida, que en definitiva será nuestra nueva capa de neuronas ocultas. Se aplican varios kernel para sacar diferentes características.

Al aplicar una convolucion, obtenemos varios mapas de características, y si quisiéramos aplicar otra capa de convolución, ésta requeriría un poder computacional importante. Por ello y para reducir el tamaño de la próxima capa una de las técnicas mas comunes es hacer un muestreo preservando las características más importantes que detectó cada filtro. Éste muestreo es conocido como **pooling**. Dentro de la red se realizan varias convoluciones y muestreos.

Una vez aplicadas las capas de convolución y muestreo deseadas, se obtienen las diferentes características de la imagen introducida. Para distinguir las diferentes clases usando estas características, se aplica un clasificador para decidir a que clase pertenece la imagen introducida en la red. Para mas información sobre la estructura de una CNN visitar [12].

Hay múltiples tipos de CNNs y clasificadores. En la Figura 2.6 vemos representada la arquitectura de una CNN junto con un clasificador. Para ver mas tipos de CNNs visitar [11].

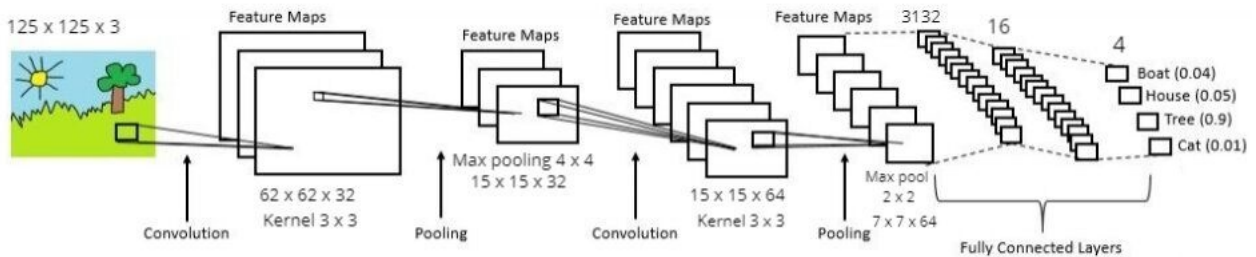


Figura 2.6: Arquitectura de una CNN junto con un clasificador

Capítulo 3

Fundamentos matemáticos del algoritmo de reducción de imagen

En este capítulo presentamos los fundamentos matemáticos expuestos en [1] que serán de gran utilidad para el desarrollo del algoritmo de reducción de imágenes. Previamente definida la idea de función de agregación en L^n , en la Sección 3.1 expondremos la noción de funciones de agregación en L^2 y mostramos un método para obtener funciones de agregación en L^2 respecto a órdenes admisibles. Posteriormente, en la Sección 3.2 mostramos resultados similares a los presentes en la Sección 3.1 pero para el caso de L^3 .

Empezamos definiendo la noción de funciones de agregación m-arias sobre el conjunto L^n .

Definición 3.1. Sean n, m enteros positivos y \leq_L un orden admisible en L^n . Una función $G_L : (L^n)^m \rightarrow L^n$ es una **función de agregación m-aria en L^n con respecto a \leq_L** , si:

$$(A1) \quad G_L(\mathbf{0}, \dots, \mathbf{0}) = \mathbf{0}$$

$$(A2) \quad G_L(\mathbf{1}, \dots, \mathbf{1}) = \mathbf{1}$$

$$(A3) \quad G_L \text{ es creciente en cada variable con respecto al orden } \leq_L$$

En las siguientes secciones nos vamos a centrar en funciones de agregación definidas en L^2 y L^3 .

3.1. Funciones de agregación en L^2

El cálculo de $G_L(X_1, \dots, X_m)$ en L^2 se puede describir de manera que usando M_1 encontramos la recta en la que se encuentra $G_L(X_1, \dots, X_m)$ y después usando M_2 decidimos en qué punto exacto de la recta se encuentra el valor de $G_L(X_1, \dots, X_m)$.

Consideremos un orden admisible \leq_M donde $\mathbf{M} = (M_1, M_2)$ con $M_1(X) = \alpha_1 x_1 + (1 - \alpha_1)x_2$ y $M_2(X) = \alpha_2 x_1 + (1 - \alpha_2)x_2$ para todo $X \in L^2$ y $\alpha_1, \alpha_2 \in L$ tal que $\alpha_1 \neq \alpha_2$. El hecho de elegir funciones de agregación cuyas componentes son medias aritméticas ponderadas permite darles una interpretación geométrica que pasamos a detallar a continuación.

Fijado $r \in L$, observamos que cada $X \in L^2$ tal que $M_1(X) = r$ se encuentra en un segmento dentro del cuadrado L^2 determinado por el punto (r, r) y paralelo al segmento $(1, 0)(\alpha_1, \alpha_1)$, tal y como puede ver en la Figura 3.1.

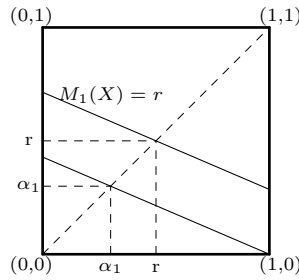


Figura 3.1: $M_1(X) = r$; donde $r = 0.5$ y $\alpha_1 = 0.3$

Sean $X, Y \in L^2$ tales que $X <_M Y$. Entonces, existen dos posibilidades:

1. $M_1(X) < M_1(Y)$, que significa que X, Y se encuentran en diferentes segmentos paralelos (determinados por las ecuaciones $M_1(X) = r_1$ y $M_1(Y) = r_2$) donde el segmento con Y se encuentra más cerca de $(1,1)$ que el segmento con X ya que $r_1 < r_2$. En la Figura 3.2 puede verse la situación antes descrita para $r_1 = 0.5$ (rojo) y $r_2 = 0.6$ (azul).

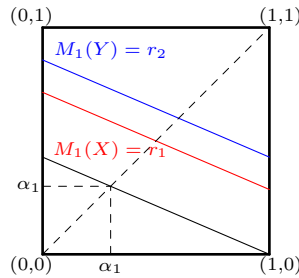


Figura 3.2: $M_1(X) < M_1(Y)$, donde $r_1 = 0.5$ y $r_2 = 0.6$

2. $M_1(X) = M_1(Y)$ y $M_2(X) < M_2(Y)$, que significa que X, Y están en el mismo segmento determinado por la ecuación $M_1(X) = M_1(Y) = r$, mientras que de la desigualdad $M_2(X) < M_2(Y)$ se sigue que:
 - Y está más cerca de $(1,0)$ que X si $\alpha_2 > \alpha_1$. En la Figura 3.3 (a) puede verse esta situación para $\alpha_1 = 0.3$ y $\alpha_2 = 0.5$.
 - X está más cerca de $(1,0)$ que Y si $\alpha_2 < \alpha_1$. En la Figura 3.3 (b) puede verse esta situación para $\alpha_1 = 0.4$ y $\alpha_2 = 0.2$.

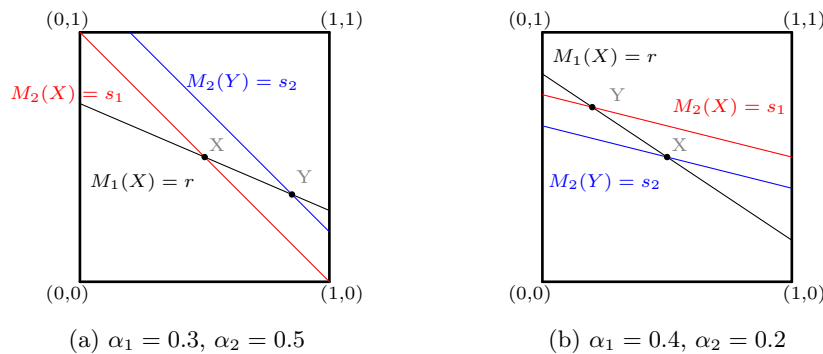


Figura 3.3: $M_1(X) = M_1(Y) = 0.5, M_2(X) = 0.5$ y $M_2(Y) = 0.6$

El siguiente resultado es un método que permite construir funciones de agregación m -arias en L^2 respecto a un orden admisible en L^2 partiendo de G_1, G_2 dos funciones de agregación m -arias en L .

Teorema 3.1. Sea m un entero positivo, sean $\alpha_1, \alpha_2 \in L$ tal que $\alpha_1 \neq \alpha_2$ y $\mathbf{M} = (M_1, M_2)$ es una dupla de funciones de agregación binarias con $M_i(X) = \alpha_i x_1 + (1 - \alpha_i)x_2$ para todo $X \in L^2$ e $i \in \{1, 2\}$. Sean G_1, G_2 funciones de agregación m -arias en L donde G_1 es estrictamente creciente. Entonces, la función $G_L : (L^2)^m \rightarrow L^2$, dada por

$$G_L(X_1, \dots, X_m) = \begin{cases} (1 - g_2)U + g_2V & , si \quad \alpha_1 > \alpha_2 \\ g_2U + (1 - g_2)V & , si \quad \alpha_1 < \alpha_2 \end{cases} \quad (3.1)$$

con $X_1, X_2, \dots, X_m \in L^2$, es una función de agregación m -aria en L^2 con respecto a un orden admisible \leq_M inducido por M , donde $g_i = G_i(M_i(X_1), \dots, M_i(X_m))$ para todo $i \in \{1, 2\}$ y

$$U = \begin{cases} (\frac{g_1}{\alpha_1}, 0) & , si \quad \alpha_1 \geq g_1 \\ (1, \frac{g_1 - \alpha_1}{1 - \alpha_1}) & , si \quad \alpha_1 < g_1 \end{cases}, V = \begin{cases} (\frac{g_1 + \alpha_1 - 1}{\alpha_1}, 1) & , si \quad \alpha_1 \geq g_1 \\ (0, \frac{g_1}{1 - \alpha_1}) & , si \quad \alpha_1 < g_1. \end{cases}$$

Por convenio, establecemos $\frac{0}{0} = 1$

Demostración. Observamos que, dado que $g_2 \in L$ y $U, V \in L^2$ para todo $X_1, \dots, X_m \in L^2$ y $\alpha_1, \alpha_2 \in L$ con $\alpha_1 \neq \alpha_2$, resulta que G_L está bien definida y por lo tanto su imagen está siempre en L^2 .

A continuación, vamos a ver que se verifican los tres axiomas presentes en la Definición 3.1

Por un lado, si $X_1 = \dots = X_m = \mathbf{0}$ resulta que $g_1 = g_2 = 0$, por lo tanto

$$G_L(\mathbf{0}, \dots, \mathbf{0}) = \begin{cases} U & , si \quad \alpha_1 > \alpha_2 \\ V & , si \quad \alpha_1 < \alpha_2 \end{cases}$$

Ya que $U = \mathbf{0}$ para todo $\alpha_1 > 0$ mientras que $V = \mathbf{0}$ para todo $\alpha_1 < 1$, el axioma (A1) queda probado.

Por otro lado, si $X_1 = \dots = X_m = \mathbf{1}$ resulta que $g_1 = g_2 = 1$, por lo tanto

$$G_L(\mathbf{1}, \dots, \mathbf{1}) = \begin{cases} V & , si \quad \alpha_1 > \alpha_2 \\ U & , si \quad \alpha_1 < \alpha_2 \end{cases}$$

y el axioma (A2) se sigue de la observación de que $U = \mathbf{1}$ para todo $\alpha_1 < 1$ mientras que $V = \mathbf{1}$ para todo $\alpha_1 > 0$

Respecto a (A3), vamos a denotar $G_L(X_1, \dots, X_m) := Z$. Entonces $g_1 = M_1(Z)$, ya que:

- De la ecuación (3.1) se tiene que Z está en el segmento UV dentro del cuadrado L^2
- Además, U, V son las intersecciones de la recta $\alpha_1 z_1 + (1 - \alpha_1)z_2 = g_1$ con los lados del cuadrado L^2 . En particular U es la intersección de la recta $\alpha_1 z_1 + (1 - \alpha_1)z_2 = g_1$ con la serie de segmentos conectados $(0, 0) - (1, 0) - (1, 1)$ y V es la intersección de la misma recta con la serie conectada de segmentos $(0, 0) - (0, 1) - (1, 1)$.

Por último, para demostrar que G_L es creciente en cada variable respecto al orden \leq_M , supongamos que $X_k^* \geq_M X_k$ para algún $k \in \{1, \dots, m\}$ y denotamos $G_L(X_1, \dots, X_{k-1}, X_k^*, X_{k+1}, \dots, X_m) := Z^*$. Ahora bien, atendiendo a la interpretación geométrica vista anteriormente hay dos posibilidades:

1. Si $X_k^* = X_k$, entonces es claro que $G_L(X_1, \dots, X_k^*, \dots, X_n) = G_L(X_1, \dots, X_k, \dots, X_n)$, con lo que $Z = Z^*$.

2. Si $X_k^* > X_k$ puede ocurrir que:

2.1. $M_1(X_k^*) > M_1(X_k)$, y usando la hipótesis de que G_1 es estrictamente creciente se tiene que

$$g_1^* = G_1(M_1(X_1), \dots, M_1(X_{k-1}), M_1(X_k^*), M_1(X_{k+1}), \dots, M_1(X_m)) \\ > G_1(M_1(X_1), \dots, M_1(X_m)) = g_1,$$

con lo que el segmento determinado por g_1^* está más cerca de $(1, 1)$ que el segmento determinado por g_1 . Por consiguiente, $Z^* >_M Z$

2.2. $M_1(X_k^*) = M_1(X_k)$ y $M_2(X_k^*) > M_2(X_k)$, donde la igualdad implica que Z^* y Z están en el mismo segmento UV . Por otro lado, de la desigualdad

$$g_2^* = G_1(M_2(X_1), \dots, M_2(X_{k-1}), M_2(X_k^*), M_2(X_{k+1}), \dots, M_2(X_m)) \\ \geq G_1(M_2(X_1), \dots, M_2(X_m)) = g_2,$$

se tiene que:

- (i) si $\alpha_2 > \alpha_1$, entonces Z^* está más cerca (o igual de cerca) a U , y en consecuencia a $(1, 0)$ que Z , ya que $Z^* >_M Z$;
- (ii) si $\alpha_2 < \alpha_1$, entonces Z está más cerca (o igual de cerca) a U , y en consecuencia a $(1, 0)$ que Z^* , ya que $Z^* >_M Z$;

Luego, en cualquier caso, se sigue que $Z^* \geq_M Z$ □

Observación 3.1. (i) Cabe señalar que si G_2 es estrictamente creciente, entonces también el crecimiento de G_L dado en el axioma (A3) es estricto.

(ii) Si $G_1 = G_2$, entonces G_L puede ser considerada (en casos concretos que veremos más adelante) como una generalización de la función de agregación G_1 de L a L^n

A continuación, presentamos un ejemplo de la función G_L descrita en el Teorema 3.1.

Ejemplo 3.1. Sean $G_1 = G_2$ medias aritméticas y $\alpha_1 = 1/2$, $\alpha_2 = 2/3$.

- a) Sean $X_1 = (0.2, 0.6)$, $X_2 = (0.1, 0.9)$, $X_3 = (0.9, 0.3)$, entonces $g_1 = 0.5$, $g_2 = 7/15$, $U = (1, 0)$, $V = (0, 1)$ y $G_L = (7/15, 8/15)$
- b) $X_1 = (0.1, 0.1)$, $X_2 = (0.3, 0.3)$, $X_3 = (0.8, 0.8)$, entonces $g_1 = 0.4$, $g_2 = 0.4$, $U = (0, 0.8)$, $V = (0.8, 0)$ y $G_L = (0.32, 0.48)$
- c) $X_1 = (0.2, 0.2)$, $X_2 = (0.4, 0.4)$, $X_3 = (0.9, 0.9)$, entonces $g_1 = 0.5$, $g_2 = 0.5$, $U = (1, 0)$, $V = (0, 1)$ y $G_L = (0.5, 0.5)$

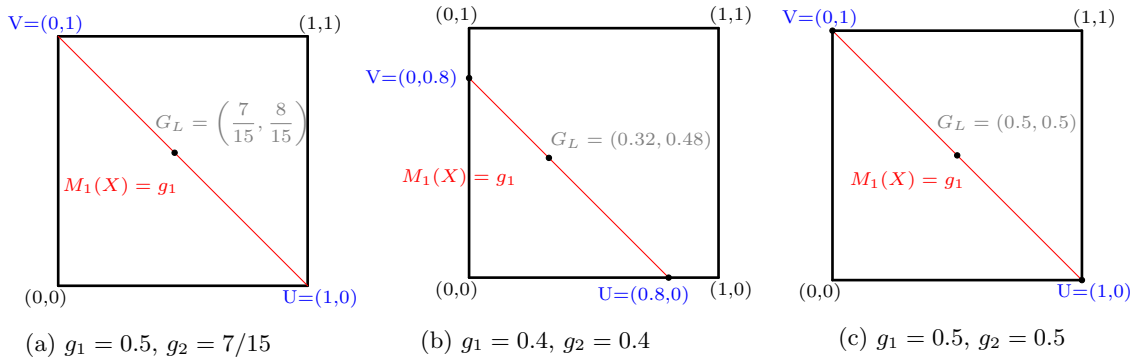


Figura 3.4: Ejemplo de función de agregación en distintos puntos

Observación 3.2. Del ejemplo anterior se tiene que G_L no es una generalización de G_1 de la forma

$$G_L((x_1, x_1), \dots, (x_m, x_m)) = (G_1(x_1, \dots, x_m), G_1(x_1, \dots, x_m))$$

para todo $x_1, \dots, x_m \in L$. De hecho, la igualdad solo se da para x_1, \dots, x_m si:

$$G_L((x_1, x_1), \dots, (x_m, x_m)) \in H$$

donde H es el conjunto:

- $H = \left\{ (z_1, z_2) \in L^2 : \frac{\alpha_1}{1-\alpha_1} - \frac{\alpha_1}{1-\alpha_1} z_1 \leq z_2 \leq 1 - \frac{\alpha_1}{1-\alpha_1} z_1, z_1 \in L \right\}$, si $\alpha_1 \leq \frac{1}{2}$
- $H = \left\{ (z_1, z_2) \in L^2 : \frac{1-\alpha_1}{\alpha_1} - \frac{1-\alpha_1}{\alpha_1} z_2 \leq z_1 \leq 1 - \frac{1-\alpha_1}{\alpha_1} z_2, z_2 \in L \right\}$, si $\alpha_1 \geq \frac{1}{2}$

El conjunto H para distintos valores de α_1 puede verse en la Figura 3.5.

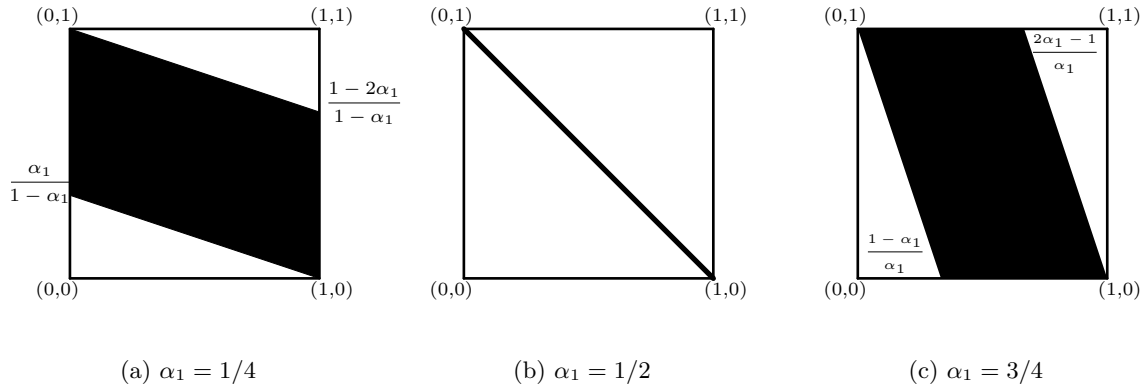


Figura 3.5: El conjunto H para valores $\alpha < 1/2$, $\alpha = 1/2$ y $\alpha > 1/2$

En la Observación 3.2, para los casos extremos de $\alpha_1 = 0$ ó $\alpha_1 = 1$ se tiene que $H = L^2$ y lo que se obtiene es el siguiente resultado.

Teorema 3.2. Bajo las hipótesis del Teorema 3.1, sea $\alpha_1 \in \{0, 1\}$. Entonces

$$G_L((x_1, x_1), \dots, (x_m, x_m)) = (G_1(x_1, \dots, x_m), G_1(x_1, \dots, x_m)) \quad (3.2)$$

para todo $x_1, \dots, x_m \in L$, si y solo si, $G_2 = G_1$

Demostración. Si $\alpha_1 = 0$, entonces para todo $x_1, \dots, x_m \in L$, tenemos que $U = (1, g_1)$ y $V = (0, g_1)$, con lo que

$$G_L((x_1, x_1), \dots, (x_m, x_m)) = g_2 U + (1 - g_2) V = (g_2, g_1)$$

Por lo tanto, la Ecuación 3.2 se cumple si y solo si $G_2(x_1, \dots, x_m) = G_1(x_1, \dots, x_m)$ para todo $x_1, \dots, x_m \in L$.

La prueba para $\alpha_1 = 1$ se sigue de manera similar observando que para todo $x_1, \dots, x_m \in L$ tenemos que $U = (g_1, 0)$ y $V = (g_1, 1)$ y

$$G_L((x_1, x_1), \dots, (x_m, x_m)) = (1 - g_2) U + g_2 V = (g_2, g_1)$$

□

La situación con la idempotencia de G_L es similar al resultado anterior. Otra vez, el conjunto H es importante ya que la propiedad que caracteriza la idempotencia se satisface solo para los $X \in H$ con algunas condiciones adicionales.

Teorema 3.3. Bajo las suposiciones del Teorema 3.1, sea $\alpha_1 \in \{0, 1\}$ y G_1, G_2 idempotentes. Entonces

$$G_L(X, \dots, X) = X$$

para todo $X \in L^2$, si y solo si, $\alpha_2 = 1 - \alpha_1$

Demostración. Sea $\alpha_1 = 0$. Observamos que para todo $X = (x_1, x_2)$, se tiene que $g_1 = M_1(X) = x_2, U = (1, x_2)$ y $V = (0, x_2)$, con lo que,

$$G_L(X, \dots, X) = g_2U + (1 - g_2)V = (g_2, x_2)$$

Luego $G_L(X, \dots, X) = X$ si y solo si $\alpha_2 = 1$

La prueba para $\alpha_1 = 1$ se sigue de manera similar observando que para todo $X = (x_1, x_2) \in L^2$, tenemos que $g_1 = M_1(X) = x_1, U = (X_1, 0)$ y $V = (X - 1, 1)$, con lo que,

$$G_L(X, \dots, X) = (1 - g_2)U + g_2V = (x_1, g_2)$$

□

En la Sección 3.2 se presentan resultados similares a los vistos en la Sección 3.1 pero adaptados al contexto de funciones de agregación en el conjunto L^3 . Mencionar que la adaptación en este contexto no es directa y requiere de un resultado previo que veremos en dicha sección.

3.2. Funciones de agregación en L^3

Esta sección nos llevará a obtener un algoritmo que implemente la construcción de una función de agregación en L^3 para usarlo más adelante en crear un nuevo método de reducción de imagen.

El cálculo de $G_L(X_1, \dots, X_m)$ en L^3 puede ser descrito de manera que usando M_1 encontramos un plano en donde $G_L(X_1, \dots, X_m)$ está (éste es el plano $M_1(X) = g_1$), utilizando M_2 encontramos la línea (que es paralela a la intersección de los planos $M_1(X) = g_1$ y $M_2(X) = g_2$) donde se encuentra $G_L(X_1, \dots, X_m)$ y finalmente con M_3 vemos que punto de la línea es $G_L(X_1, \dots, X_m)$.

Al igual que ocurre en la Sección 3.1., la elección de funciones de agregación cuyas componentes son medias aritméticas ponderadas permite darles una interpretación geométrica que pasamos a detallar a continuación.

Tal y como se hizo en la Sección 3.1, describamos la interpretación geométrica de los órdenes admisibles \leq_M en L^3 , donde $\mathbf{M} = (M_1, M_2, M_3)$ con $M_i(X) = \alpha_{i1}x_1 + \alpha_{i2}x_2 + \alpha_{i3}x_3$, para $i \in \{1, 2, 3\}$ y $(\alpha_{i1}, \alpha_{i2}, \alpha_{i3}) \in L^3$ son vectores linealmente independientes tal que $\alpha_{i1} + \alpha_{i2} + \alpha_{i3} = 1$, para todo $i \in \{1, 2, 3\}$. Sean $X, Y \in L^3$ tales que $X <_M Y$. Entonces, existen tres posibilidades:

1. $M_1(X) < M_1(Y)$, lo que significa que X, Y están en diferentes planos paralelos (determinados por las ecuaciones $M_1(X) = r_1$ y $M_1(Y) = r_2$) donde el plano que contiene a Y está mas cerca a $(1, 1, 1)$ que el plano que contiene a X ya que $r_2 > r_1$.
2. $M_1(X) = M_1(Y)$ y $M_2(X) < M_2(Y)$, lo que significa que X, Y están en el mismo plano $M_1(X) = r_1$, pero en diferentes planos paralelos determinados por las ecuaciones $M_2(X) = r_3$ y $M_2(Y) = r_4$, donde el plano que contiene a Y está mas cerca a $(1, 1, 1)$ que el plano que contiene a X ya que $r_4 > r_3$. En otras palabras, X, Y están en diferentes segmentos paralelos en el plano $M_1(X) = r_1$.

3. $M_1(X) = M_1(Y)$, $M_2(X) = M_2(Y)$ y $M_3(X) < M_3(Y)$, lo que significa que X, Y están en el mismo segmento dentro de L^3 determinado por la intersección de los planos $M_1(X) = r_1$ y $M_2(X) = r_2$, pero en diferentes planos paralelos determinados por las ecuaciones $M_3(X) = r_3$ y $M_3(Y) = r_4$, donde el plano que contiene a Y está más cerca a $(1, 1, 1)$ que el plano que contiene a X ya que $r_4 > r_3$. (c)

A continuación, veamos un ejemplo ilustrativo que permita visualizar las distintas posibilidades para órdenes admisibles en L^3 que hemos descrito anteriormente.

Ejemplo 3.2. En el conjunto L^3 sea \mathbf{M} el orden inducido por la función de agregación $\mathbf{M} = (M_1, M_2, M_3)$ donde

- $M_1(X) = 0.2x_1 + 0.4x_2 + 0.4x_3$
- $M_2(X) = 0.5x_1 + 0.2x_2 + 0.3x_3$
- $M_3(X) = 0.4x_1 + 0.6x_3$,

observamos gráficamente en la Figura 3.6 (a), que para ciertos $X, Y \in L^3$, $M_1(X) = 0.4$ (plano azul) y $M_1(Y) = 0.7$ (plano rojo), y por lo tanto $M_1(X) \leq M_1(Y)$.

Además en la Figura 3.6 (b), observamos que para ciertos $X, Y \in L^3$, $M_1(X) = M_1(Y) = 0.4$ (plano azul), $M_2(X) = 0.3$ (recta roja) y $M_2(Y) = 0.7$ (recta amarilla), y por lo tanto $M_2(X) \leq M_2(Y)$.

Y finalmente en la Figura 3.6 (c), observamos que para ciertos $X, Y \in L^3$, $M_1(X) = M_1(Y) = 0.4$ (plano azul), $M_2(X) = M_2(Y) = 0.3$ (recta roja), $M_3(X) = 0.3$ (punto rosa) y $M_3(Y) = 0.6$ (punto marrón), y por lo tanto $M_3(X) \leq M_3(Y)$.

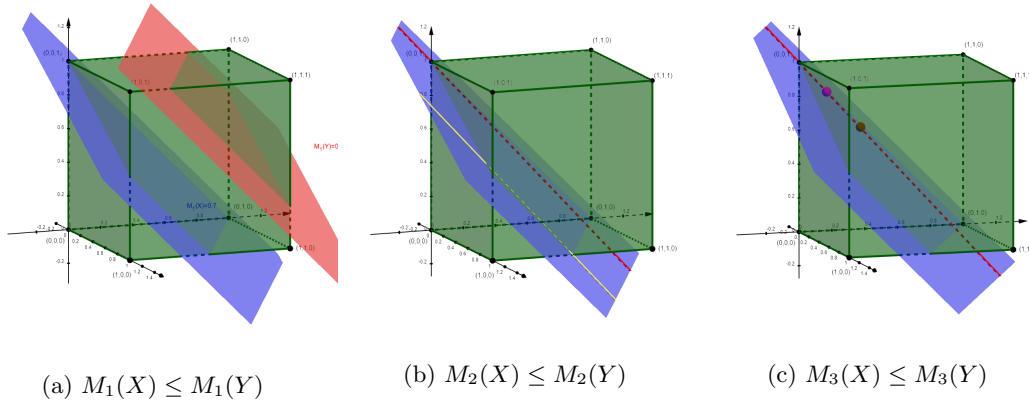


Figura 3.6: Interpretación geométrica del orden \mathbf{M} del Ejemplo 3.2

Llegados a este punto, para poder presentar el teorema equivalente al Teorema 3.1 en L^3 necesitamos el siguiente resultado previo.

Proposición 3.1. Sean $g_1, g_2, g_3 \in L$, sean $(\alpha_{i1}, \alpha_{i2}, \alpha_{i3}) \in L^3$ vectores linealmente independientes tal que $\alpha_{i1} + \alpha_{i2} + \alpha_{i3} = 1$, para $i \in \{1, 2, 3\}$ y sea M_1, M_2, M_3 una terna de funciones de agregación en L con $M_i(X) = \alpha_{i1}x_1 + \alpha_{i2}x_2 + \alpha_{i3}x_3$ para todo, $X = (x_1, x_2, x_3) \in L^3$ e $i \in \{1, 2, 3\}$. Entonces:

- (i) Existe exactamente una única recta l_1 que es la intersección de los planos $M_1(x) = g_1$ y $M_2(X) = g_2$
- (ii) Sea $\mathcal{A} = \{X = (x_1, x_2, x_3) \in L^3 \mid M_1(X) = g_1 \text{ y } x_i \in \{0, 1\} \text{ para algún } i \in \{1, 2, 3\}\}$. Entonces \mathcal{A} es no vacío. Además, si $M_2(Y) = M_2(Z)$ para algunos $Y, Z \in \mathcal{A}$ tal que $Y \neq Z$, entonces la recta determinada por los puntos Y, Z es paralela a l_1
- (iii) Existen U_2 y $V_2 \in \mathcal{A}$ tal que $M_2(U_2) \leq M_2(X)$ y $M_2(V_2) \geq M_2(X)$ para todo $X \in \mathcal{A}$
- (iv) Existe exactamente una recta l_2 , contenida en el plano $M_1(X) = g_1$, l_2 es paralela a la recta l_1 y el punto $(1 - g_2)U_2 + g_2V_2$ está en l_2
- (v) Existe exactamente un $U_3 \in l_2 \cap \mathcal{A}$ y exactamente un $V_3 \in l_2 \cap \mathcal{A}$ tal que $M_3(U_3) \leq M_3(X)$ y $M_3(V_3) \geq M_3(X)$ para todo $X \in l_2 \cap \mathcal{A}$

Demostración. (i) Observamos que $(\alpha_{11}, \alpha_{12}, \alpha_{13})$ y $(\alpha_{21}, \alpha_{22}, \alpha_{23})$ son vectores normales de los planos $M_1(X) = g_1$ y $M_2(X) = g_2$, respectivamente. Como los vectores son linealmente independientes, los planos no son paralelos.

(ii) Como (g_1, g_1, g_1) está en L^3 y además $M_1(X) = g_1$, entonces dicho plano $M_1(X) = g_1$ tiene intersección no vacía con, al menos, uno de los lados del cubo L^3 . Por lo tanto, \mathcal{A} es no vacío. Además, si $Y \neq Z$ está en el plano $M_1(X) = g_1$ y $M_2(X) = M_2(Z)$, entonces tenemos que los puntos Y, Z se encuentran en el mismo plano paralelo al plano $M_2(X) = g_2$ y en consecuencia la recta determinada por los puntos Y, Z es paralela a l_1 .

(iii) Se da directamente con la definición del conjunto \mathcal{A} ya que como hemos probado en (ii), \mathcal{A} es no vacío.

(iv) Se sigue del hecho de que el punto $(1 - g_2)U_2 + g_2V_2$ está en el plano $M_1(X) = g_1$

(v) Observamos que l_2 es una recta que se encuentra en el plano $M_1(X) = g_1$ y es paralela al plano $M_2(X) = g_2$, por lo tanto como los vectores $(\alpha_{11}, \alpha_{12}, \alpha_{13})$, $(\alpha_{21}, \alpha_{22}, \alpha_{23})$ y $(\alpha_{31}, \alpha_{32}, \alpha_{33})$ son linealmente independientes, l_2 no es paralela al plano $M_3(X) = g_3$. Esto significa que para todo $Y, Z \in L^2$ se tiene que $M_3(Y) \neq M_3(Z)$ \square

La proposición anterior nos da una construcción única de los puntos U_3 y V_3 con la que poder definir, en la línea del Teorema 3.1, una función de agregación m-aria $G_L : (L^3)^m \rightarrow L^3$ con respecto a un orden admisible en L^3 .

El siguiente Teorema es el que nos va a permitir desarrollar el algoritmo que usaremos para el método de reducción de imagen.

Teorema 3.4. Sea m un entero positivo, sea $\alpha_{i1} + \alpha_{i2} + \alpha_{i3} = 1$, para todo $i \in \{1, 2, 3\}$ y sea $\mathbf{M} = (M_1, M_2, M_3)$ una terna de funciones de agregación en L con $M_i(X) = \alpha_{i1}x_1 + \alpha_{i2}x_2 + \alpha_{i3}x_3$, para todo $X = (x_1, x_2, x_3) \in L^3$, e, $i \in \{1, 2, 3\}$. Sea $\mathbf{G} = (G_1, G_2, G_3)$ una secuencia de de funciones de agregación m-arias en L donde G_1 y G_2 son estrictamente crecientes. Entonces la función $G_L : (L^3)^m \rightarrow L^3$, para todo $X_1, X_2, \dots, X_m \in L^3$ dada por:

$$G_L(X_1, \dots, X_m) = (1 - g_3)U_3 + g_3V_3 \quad (3.3)$$

es una función de agregación m-aria en L^3 con respecto al orden admisible \leq_M inducido por \mathbf{M} , siendo $g_i = G_i(M_i(X_1), \dots, M_i(X_m))$ para todo $i \in \{1, 2, 3\}$ y donde U_3 y V_3 vienen dados por la Proposición 3.1 (v).

Demostración. Observamos que, como $g_3 \in L$ y $U_3, V_3 \in L^3$ para todo $X_1, \dots, X_m \in L^3$, tenemos que G_L está bien definida, por lo tanto sus imágenes están siempre en L^3 .

A continuación, vamos a ver que se verifican los tres axiomas presentes en la Definición 3.1.

(A1) Sean $X_1 = \dots, X_m = \mathbf{0}$. Entonces $g_1 = g_2 = g_3 = 0$ y por la Proposición 3.1 se tiene que (i) $\mathbf{0} \in l_1$, (ii) $\mathbf{0} \in \mathcal{A}$, (iii) $U_2 = \mathbf{0}$, (iv) $\mathbf{0} \in l_2$ y (v) $U_3 = \mathbf{0}$. Por lo tanto $G_L(\mathbf{0}, \dots, \mathbf{0}) = \mathbf{0}$

(A2) Sean $X_1 = \dots, X_m = \mathbf{1}$. Entonces $g_1 = g_2 = g_3 = 1$ y por la Proposición 3.1 se tiene que (i) $\mathbf{1} \in l_1$, (ii) $\mathbf{1} \in \mathcal{A}$, (iii) $V_2 = \mathbf{1}$, (iv) $\mathbf{1} \in l_2$ y (v) $V_3 = \mathbf{1}$. Por lo tanto $G_L(\mathbf{1}, \dots, \mathbf{1}) = \mathbf{1}$

(A3) Respecto a que G_L es creciente en cada componente, supongamos que $X_k^* \geq_{\mathbf{M}} X_k$ para algún $k \in \{1, \dots, m\}$ y denotamos $G_L(X_1, \dots, X_m) = Z$ y $G_L(X_1, \dots, X_{k-1}, X_k^*, X_{k+1}, \dots, X_m) = Z^*$. Desde ahora, con el índice $*$ nos referimos a los términos correspondientes a Z^* . Entonces existen las siguientes posibilidades:

1. Si $X_k^* = X_k$, entonces es claro que $G_L(X_1, \dots, X_k^*, \dots, X_m) = G_L(X_1, \dots, X_k, \dots, X_m)$, con lo que $Z = Z^*$.
2. Si $X_k^* > X_k$ puede ocurrir que:
 - (i) $M_1(X_k^*) > M_1(X_k)$, de donde tenemos que

$$g_1^* = G_1(M_1(X_1), \dots, M_1(X_{k-1}), M_1(X_k^*), M_1(X_{k+1}), \dots, M_1(X_m)) > \\ G_1(M_1(X_1), \dots, M_1(X_m)) = g_1$$

Como $U_3, V_3 \in \mathcal{A}$ y $U_3^*, V_3^* \in \mathcal{A}^* \neq \mathcal{A}$ (al ser $g_1^* > g_1$), tenemos que

$$M_1(U_3^*) = M_1(V_3^*) = g_1^* > g_1 = M_1(U_3) = M_1(V_3)$$

Por lo tanto,

$$M_1((1 - g_3^*)U_3^* + g_3^*V_3^*) = (1 - g_3^*)M_1(U_3^*) + g_3^*M_1(V_3^*) > \\ (1 - g_3)M_1(U_3) + g_3M_1(V_3) = M_1((1 - g_3)U_3 + g_3V_3)$$

y en consecuencia $Z^* >_{\mathbf{M}} Z$.

- (ii) $M_1(X_k^*) = M_1(X_k)$ y $M_2(X_k^*) > M_2(X_k)$, de la igualdad se sigue que $g_1^* = g_1$ mientras que de la desigualdad se tiene que $g_2^* > g_2$. Entonces, dado que $U_3, V_3, U_3^*, V_3^* \in \mathcal{A}^* = \mathcal{A}$ (al ser $g_1^* = g_1$) se sigue que

$$M_1((1 - g_3^*)U_3^* + g_3^*V_3^*) = M_1((1 - g_3)U_3 + g_3V_3)$$

Además, l_1^* es paralela a l_1 y $U_2^* = U_2, V_2^* = V_2$, así la recta l_2^* está mas cerca del punto V_2 que la recta l_2 , y en consecuencia $M_2(X^*) > M_2(X)$ para todo $X^* \in l_2^*$ y $X \in l_2$. Finalmente, como $U_3, V_3 \in l_2$ y $U_3^*, V_3^* \in l_2^*$, tenemos que

$$M_2((1 - g_3^*)U_3^* + g_3^*V_3^*) > M_2((1 - g_3)U_3 + g_3V_3)$$

y en consecuencia $Z^* >_{\mathbf{M}} Z$.

- (iii) $M_1(X_k^*) = M_1(X_k), M_2(X_k^*) = M_2(X_k)$ y $M_3(X_k^*) > M_3(X_k)$, de donde de la primera igualdad se sigue que $g_1^* = g_1$, de la segunda igualdad se sigue que $g_2^* = g_2$ mientras que de la desigualdad se tiene que $g_3^* > g_3$. Así $U_3^* = U_3$ y $V_3^* = V_3$, que significa que el punto $(1 - g_3^*)U_3^* + g_3^*V_3^*$ está mas cerca del punto V_3 que el punto $(1 - g_3)U_3 + g_3V_3$. Entonces,

$$M_1((1 - g_3^*)U_3^* + g_3^*V_3^*) = M_1((1 - g_3)U_3 + g_3V_3)$$

$$M_2((1 - g_3^*)U_3^* + g_3^*V_3^*) = M_2((1 - g_3)U_3 + g_3V_3)$$

$$M_3((1 - g_3^*)U_3^* + g_3^*V_3^*) > M_3((1 - g_3)U_3 + g_3V_3)$$

y finalmente $Z^* >_{\mathbf{M}} Z$.

Luego, en cualquier caso, se sigue que $Z^* \geq_M Z$ \square

Observación 3.3. (i) Al igual que ocurre para el caso de L^2 , si G_3 es estrictamente creciente, entonces también el crecimiento de G_L dado en el axioma (A3) es estricto. (ii) Si $G_1 = G_2 = G_3$, entonces G_L puede ser considerada (en casos concretos que veremos más adelante) como una generalización de la función de agregación G_1 de L a L^n

3.3. Construcción de funciones de agregación respecto a un orden admisible

A continuación, presentamos un algoritmo que nos permite calcular de forma sistemática el valor de la función de agregación m -aria G_L presente en el Teorema 3.4. El desarrollo de este algoritmo es necesario ya que involucra pasos intermedios para calcular los valores U_2, V_2, U_3 y V_3 presentes en la Proposición 3.1.

Inputs:

- $X_1, \dots, X_m \in L^3$;
- Vectores linealmente independientes $(\alpha_{i1}, \alpha_{i2}, \alpha_{i3}) \in L^3$ tal que $\alpha_{i1} + \alpha_{i2} + \alpha_{i3} = 1$ para todo $i \in \{1, 2, 3\}$. Los representaremos en forma matricial

$$A_M = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix}.$$

- Dadas G_1, G_2 y G_3 funciones de agregación m -arias en L , con G_1 y G_2 estrictamente crecientes, definimos la terna de funciones de agregación m -arias $\mathbf{G} = (G_1, G_2, G_3)$

Algoritmo:

0. Comprobamos que $\det(A_M) \neq 0$ y que las filas de A_M sumen 1 para asegurar que el orden \leq_M es admisible, según la Proposición 2.2
1. Calculamos $g_i = G_i(M_i(X_1), \dots, M_i(X_m))$ para todo $i \in \{1, 2, 3\}$ donde M_i es la fila i -ésima de la matriz A_M
2. Calculamos el producto vectorial de M_1 y M_2 y lo denotamos como $Z = (z_1, z_2, z_3)$
3. Encontramos el mínimo conjunto \mathcal{P}_1 (el conjunto de todos los puntos de la intersección entre el plano $M_1(X) = g_1$ y los bordes del cubo $[0, 1]^3$ que no son paralelos a Z considerado como un vector) tal que:
 - si $\alpha_{13} \neq 0$, entonces $\left\{ (0, 0, \frac{g_1}{\alpha_{13}}), (0, 1, \frac{g_1 - \alpha_{12}}{\alpha_{13}}), (1, 0, \frac{g_1 - \alpha_{11}}{\alpha_{13}}), (1, 1, \frac{g_1 - \alpha_{11} - \alpha_{12}}{\alpha_{13}}) \right\} \cap L^3 \subseteq \mathcal{P}_1$;
 - si $\alpha_{12} \neq 0$, entonces $\left\{ (0, \frac{g_1}{\alpha_{12}}, 0), (0, \frac{g_1 - \alpha_{13}}{\alpha_{12}}, 1), (1, \frac{g_1 - \alpha_{11}}{\alpha_{12}}, 0), (1, \frac{g_1 - \alpha_{11} - \alpha_{13}}{\alpha_{12}}, 1) \right\} \cap L^3 \subseteq \mathcal{P}_1$;
 - si $\alpha_{11} \neq 0$, entonces $\left\{ (\frac{g_1}{\alpha_{11}}, 0, 0), (\frac{g_1 - \alpha_{13}}{\alpha_{11}}, 0, 1), (\frac{g_1 - \alpha_{12}}{\alpha_{11}}, 1, 0), (\frac{g_1 - \alpha_{12} - \alpha_{13}}{\alpha_{11}}, 1, 1) \right\} \cap L^3 \subseteq \mathcal{P}_1$.
4. Encontramos $U_2, V_2 \in \mathcal{P}_1$ tal que $M_2(U_2) \leq M_2(X)$ y $M_2(V_2) \geq M_2(X)$ como aplicación de la Proposición 3.1.
5. Calculamos $P := (1 - g_2)U_2 + g_2V_2$ y denotamos $P = (p_1, p_2, p_3)$

3.3. CONSTRUCCIÓN DE FUNCIONES DE AGREGACIÓN RESPECTO A UN ORDEN ADMISIBLE33

6. Encontramos el conjunto \mathcal{P}_2 (el conjunto de todos los puntos de la intersección entre la recta dada por la ecuación $X = P + tZ, t \in \mathbb{R}$, y las caras del cubo $[0, 1]^3$ que no son paralelos a Z considerado como un vector) tal que:
 - si $z_1 \neq 0$, entonces $\left\{ \left(0, \frac{p_2 z_1 - p_1 z_2}{z_1}, \frac{p_3 z_1 - p_1 z_3}{z_1} \right), \left(1, \frac{p_2 z_1 - p_1 z_2 + z_2}{z_1}, \frac{p_3 z_1 - p_1 z_3 + z_3}{z_1} \right) \right\} \cap L^3 \subseteq \mathcal{P}_2$;
 - si $z_2 \neq 0$, entonces $\left\{ \left(\frac{p_1 z_2 - p_2 z_1}{z_2}, 0, \frac{p_3 z_2 - p_2 z_3}{z_2} \right), \left(\frac{p_1 z_2 - p_2 z_1 + z_1}{z_2}, 1, \frac{p_3 z_2 - p_2 z_3 + z_3}{z_2} \right) \right\} \cap L^3 \subseteq \mathcal{P}_2$;
 - si $z_3 \neq 0$, entonces $\left\{ \left(\frac{p_1 z_3 - p_3 z_1}{z_3}, \frac{p_2 z_3 - p_3 z_2}{z_3}, 0 \right), \left(\frac{p_1 z_3 - p_3 z_1 + z_1}{z_3}, \frac{p_2 z_3 - p_3 z_2 + z_2}{z_3}, 1 \right) \right\} \cap L^3 \subseteq \mathcal{P}_2$.
7. Encontramos $U_3, V_3 \in \mathcal{P}_2$ tal que $M_3(U_3) \leq M_3(X)$ y $M_3(V_3) \geq M_3(X)$ como aplicación de la Proposición 3.1.
8. Calculamos la agregación $G_L(X_1, \dots, X_m) := (1 - g_3)U_3 + g_3V_3$

A continuación, veamos dos ejemplos ilustrativos del algoritmo anteriormente descrito.

Ejemplo 3.3. Empezamos con un ejemplo sencillo considerando las medias aritméticas tales que la matriz A_M sea la identidad.

Inputs:

- $X_1 = (0.1, 0.2, 0.3); X_2 = (0.4, 0.5, 0.6); X_3 = (0.6, 0.3, 0.2); X_4 = (0.6, 0.9, 0.1)$
- $A_M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- $G_1 = G_2 = G_3 = \text{media aritmética}$

Algoritmo:

0. Observamos que $\det(A_M) = 1$ y la suma de cada una de las filas es igual a 1
1. Calculamos

$$g = G \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.1 & 0.4 & 0.6 & 0.6 \\ 0.2 & 0.5 & 0.3 & 0.9 \\ 0.3 & 0.6 & 0.2 & 0.1 \end{pmatrix} \right) = (0.425, 0.475, 0.3)$$

2. Calculamos el producto vectorial

$$\det \begin{pmatrix} i & -j & k \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = 0i + 0j + k$$

Entonces tenemos que $Z = (0, 0, 1)$

3. Como α_{12} y α_{13} son 0 y $\alpha_{11} \neq 0$ tenemos que $\left\{ \left(\frac{0.425}{1}, 0, 0 \right), \left(\frac{0.425-0}{1}, \right), \left(\frac{0.425-0}{1}, 1, 0 \right), \left(\frac{0.425-0-0}{1}, 1, 1 \right) \right\} \cap L^3 \subseteq \mathcal{P}_1 \Rightarrow \mathcal{P}_1 = \{(0.425, 0, 0), (0.425, 0, 1), (0.425, 1, 0), (0.425, 1, 1)\}$, donde este conjunto \mathcal{P}_1 puede verse en la Figura 3.7 (a)
4. Aplicando $M_2(x_1, x_2, x_3) = g_2$ a todos los puntos de \mathcal{P}_1 obtenemos que $U_2 = (0.425, 0, 0)$ y $V_2 = (0.425, 1, 0)$, tal y como puede verse en Figura 3.7 (b)

5. Calculamos $P = (1 - 0.475)U_2 + 0.475V_2 = (0.425, 0.475, 0)$ y que hemos dibujado de color rojo en la Figura 3.7 (c)

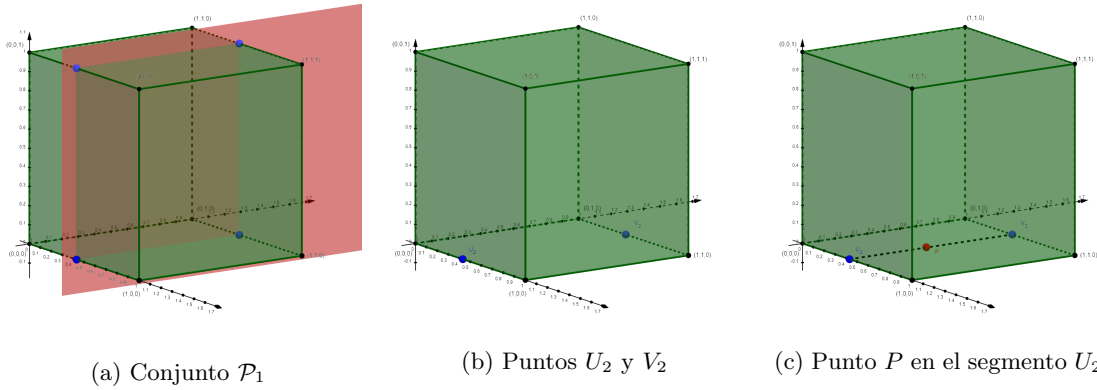


Figura 3.7: Pasos 3, 4 y 5 del algoritmo para el Ejemplo 3.3

6. Como $Z = (0, 0, 1)$, tenemos que $\left\{ \left(\frac{0.425 \cdot 1 - 0 \cdot 0}{1}, \frac{0.475 \cdot 1 - 0 \cdot 0}{1}, 0 \right), \left(\frac{0.425 \cdot 1 - 0 \cdot 0 + 0}{1}, \frac{0.475 \cdot 1 - 0 \cdot 0 + 0}{1}, 1 \right) \right\} \cap L^3 \subseteq \mathcal{P}_2 \Rightarrow \mathcal{P}_2 = \{(0.425, 0.475, 0), (0.425, 0.475, 1)\}$, tal y como puede verse en Figura 3.8 (a)
7. Aplicando $M_3(x_1, x_2, x_3) = g_3$ a todos los puntos de \mathcal{P}_2 obtenemos que: $U_3 = (0.425, 0.475, 0)$ y $V_3 = (0.425, 0.475, 1)$, tal y como puede verse en Figura 3.8 (b)
8. Finalmente $G_L(X_1, X_2, X_3, X_4) = (1 - 0.3)U_3 + 0.3V_3 = (0.425, 0.475, 0.3)$, punto que hemos señalado de color rojo en la Figura 3.8 (c)

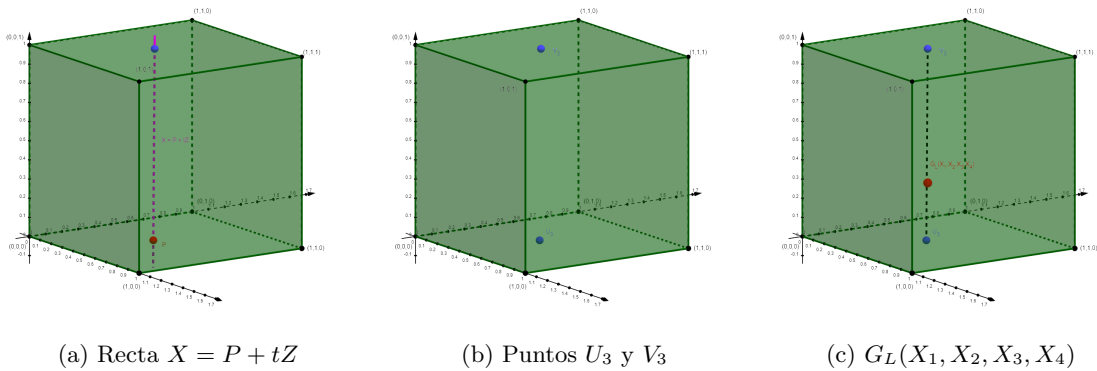


Figura 3.8: Pasos 6, 7 y 8 del algoritmo para el Ejemplo 3.3

Ejemplo 3.4. En este caso, más complejo que el anterior, consideramos medias aritméticas ponderadas tales que la matriz A_M sea regular.

Inputs:

- $X_1 = (0.1, 0.2, 0.3); X_2 = (0.4, 0.5, 0.6); X_3 = (0.6, 0.3, 0.2); X_4 = (0.6, 0.9, 0.1)$
- $A_M = \begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.5 \\ 0.5 & 0.2 & 0.3 \end{pmatrix}$

3.3. CONSTRUCCIÓN DE FUNCIONES DE AGREGACIÓN RESPECTO A UN ORDEN ADMISIBLE35

- G_1 =media aritmética; G_2 =mediana G_3 =; OWA($a = 0, b = 0,5$)

Algoritmo:

0. Observamos que $\det(A_M) = 1$ y la suma de cada una de las filas es igual a 1

1. Calculamos

$$g = G \left(\begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.5 \\ 0.5 & 0.2 & 0.3 \end{pmatrix} \begin{pmatrix} 0.1 & 0.4 & 0.6 & 0.6 \\ 0.2 & 0.5 & 0.3 & 0.9 \\ 0.3 & 0.6 & 0.2 & 0.1 \end{pmatrix} \right) = (0.39, 0.375, 0.495)$$

2. Calculamos el producto vectorial

$$\det \begin{pmatrix} i & -j & k \\ 0.3 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.5 \end{pmatrix} = 0.07i - 0.03j + -0.03k$$

Entonces tenemos que $Z = (0.07, -0.03, -0.03)$

3. Tenemos que $\mathcal{P}_1 = \{(0, 0, 0.975), (0, 1, 0.225), (1, 0, 0.225), (1, 0.3, 0), (0, 3, 1, 0)\}$, este conjunto \mathcal{P}_1 puede verse en la Figura 3.9 (a)

4. Aplicando $M_2(x_1, x_2, x_3) = g_2$ a todos los puntos de \mathcal{P}_1 obtenemos que $U_2 = (0.3, 1, 0)$ y $V_2 = (0, 0, 0.975)$, tal y como puede verse en Figura 3.9 (b)

5. Calculamos $P = (1 - 0.375)U_2 + 0.375V_2 = (0.1875, 0.625, 0.36562)$ y que hemos dibujado de color rojo en la Figura 3.9 (c)

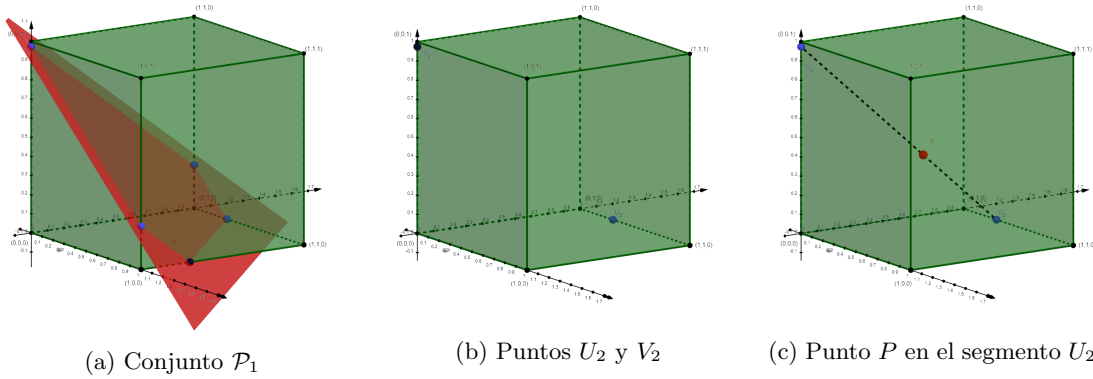


Figura 3.9: Pasos 3, 4 y 5 del algoritmo para el Ejemplo 3.4

6. Tenemos que $\mathcal{P}_2 = \{(0, 0.70536, 0.44598), (1, 0.27679, 0.01741)\}$, este conjunto \mathcal{P}_2 puede verse en la Figura 3.10 (a)

7. Aplicando $M_3(x_1, x_2, x_3) = g_3$ a todos los puntos de \mathcal{P}_2 obtenemos que $U_3 = (0, 0.70536, 0.44598)$ y $V_3 = (1, 0.27679, 0.01741)$, tal y como puede verse en Figura 3.10 (b)

8. Finalmente $G_L(X_1, X_2, X_3, X_4) = (1 - 0.495)U_3 + 0.495 \cdot V_3 = (0.495, 0.49321429, 0.23383929)$, punto que hemos señalado de color rojo en la Figura 3.10 (c)

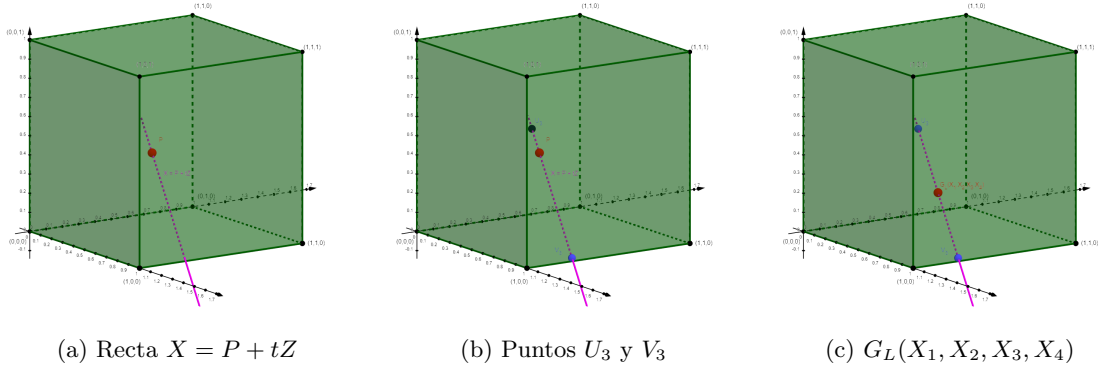


Figura 3.10: Pasos 6, 7 y 8 del algoritmo para el Ejemplo 3.4

En el siguiente resultado demostraremos la idempotencia de las funciones de agregación en L^3 , propiedad que debe cumplir $G_L(X_1, \dots, X_m)$ como vimos en el Capítulo 2.

Teorema 3.5. Bajo las suposiciones del Teorema 3.4, sea σ una permutación de $\{1, 2, 3\}$, $\alpha_{1\sigma(1)} = \alpha_{2\sigma(2)} = \alpha_{3\sigma(3)} = 1$ y sean G_1, G_2, G_3 idempotentes. Entonces G_L es idempotente.

Demostración. Sea $Y = (y_1, y_2, y_3) \in L^3$, vemos que $G_L(Y, \dots, Y) = Y$. Claramente $g_1 = y_{\sigma(1)}, g_2 = y_{\sigma(2)}$ y $g_3 = y_{\sigma(3)}$. Entonces la recta l_1 está dada por $X = (y_1, y_2, 0) + t(\alpha_{31}, \alpha_{32}, \alpha_{33}), t \in \mathbb{R}$. Por lo tanto, U_2 puede ser cualquier punto del segmento $U^d U^h$ donde $u_{\sigma(1)}^d = u_{\sigma(1)}^h = y_1, u_{\sigma(2)}^d = u_{\sigma(2)}^h = 0, u_{\sigma(3)}^d = 0, u_{\sigma(3)}^h = 1$ y V_2 puede ser cualquier punto del segmento $V^d V^h$ donde $v_{\sigma(1)}^d = v_{\sigma(1)}^h = y_1, v_{\sigma(2)}^d = v_{\sigma(2)}^h = 1, v_{\sigma(3)}^d = 0, v_{\sigma(3)}^h = 1$. Por lo tanto la recta l_2 es igual a la recta l_1 de lo que se sigue que las coordenadas de los puntos U_3, V_3 son $u_{3\sigma(1)} = v_{3\sigma(1)} = y_1, u_{3\sigma(2)} = v_{3\sigma(2)} = y_2, u_{3\sigma(3)} = 0$ y $v_{3\sigma(3)} = 1$. Entonces $G_L(Y, \dots, Y) = (1 - g_3)U_3 + g_3V_3 = (y_1, y_2, y_3) = Y$. \square

Capítulo 4

Algoritmo de reducción de imágenes y experimentación

En este capítulo vamos a ver cómo se puede adaptar el algoritmo presentado en la Sección 3.3 del Capítulo 3 para desarrollar un método de reducción de imágenes.

Como hemos comentado en el Capítulo 2, dada una imagen en mapa de bits de $m \times n$ píxeles, podemos considerar cada píxel de la imagen como el vector $i \times j \times 3$, siendo la última componente la coloración en el modelo RGB.

Con el objetivo de entender cómo funciona el algoritmo reducción de imagen, vamos a considerar la siguiente situación a pequeña escala: tenemos una imagen de cuatro píxeles con P_1, P_2, P_3 y P_4 cuyos valores de color en el modelo RGB son:

$$P_1 = (25, 51, 76), P_2 = (102, 127, 153), P_3 = (153, 76, 51), P_4 = (153, 229, 25).$$

Tras dividir cada componente de los píxeles P_1, P_2, P_3 y P_4 entre 255, obtenemos los vectores normalizados X_1, X_2, X_3 y X_4 , respectivamente, con valores en L^3 dados por:

$$X_1 = (0.1, 0.2, 0.3), X_2 = (0.4, 0.5, 0.6), X_3 = (0.6, 0.3, 0.2), X_4 = (0.6, 0.9, 0.1).$$

A modo de ilustrar del algoritmo, por comodidad hemos considerado los valores X_1, X_2, X_3 y X_4 con un decimal. Ahora bien, aplicando el algoritmo de construcción de funciones de agregación respecto a órdenes admisibles presentado en la Sección 3.3. y observando que son los mismos valores que en el Ejemplo 3.4, obtenemos el valor $X = G_L(X_1, X_2, X_3, X_4) = (0.495, 0.493, 0.234)$. Por último, multiplicando cada componente de X por 255 obtenemos el píxel $P = (126, 126, 60)$.

En resumen, hemos reducido la ventana 2x2 formada por los píxeles P_1, P_2, P_3 y P_4 en un único píxel cuyo valor es P .

El objetivo de este capítulo es reducir imágenes médicas con el algoritmo desarrollado en el Capítulo 3 y comparar el resultado con la reducción de esas mismas imágenes con los mejores métodos de la literatura. Además al final de este capítulo veremos una aplicación de esta reducción.

Como hemos comentado en la Sección 2.2.1., la reducción de imagen se lleva acabo agregando una ventana de píxeles a un único valor, haciendo esto por cada una de las ventanas y cada una de las tres capas de la imagen. El algoritmo desarrollado en el Capitulo 3 es bastante novedoso, ya que a

diferencia de esos métodos comunes, como el algoritmo es una función de agregación en L^3 , agregamos las tres capas a la vez.

Primeramente realizaremos pruebas para decidir que inputs elegir dentro del algoritmo. Para ello debemos decidir que matriz A_M y que tres agregaciones G_1, G_2, G_3 se va a escoger.

En este trabajo elegimos la matriz A_M igual a la identidad ya que al usar otra, como observamos en la Figura 4.1 al aplicar la reducción la imagen se ve alterada de manera que un color se destaca mas que otro, en mayor o menor medida, y la imagen se oscurece o aclara.



(a) Imagen original



$$(b) A_M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



$$(c) A_M = \begin{pmatrix} 1 & 0 & 0 \\ 2/3 & 1/6 & 1/6 \\ 2/3 & 1/6 & 1/6 \end{pmatrix}$$



$$(d) A_M = \begin{pmatrix} 0 & 1 & 0 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$



$$(e) A_M = \begin{pmatrix} 0 & 0 & 1 \\ 1/6 & 1/6 & 2/3 \\ 1/6 & 1/6 & 2/3 \end{pmatrix}$$

Figura 4.1: Imagen reducida con diferentes matrices A_M

Para llevar a cabo la elección de las funciones de agregación G_1, G_2 y G_3 , que devuelvan el mejor resultado, presentes en el Teorema 3.4 así como en el algoritmo de la Sección 3.3., hemos decidido realizar pruebas con las siguientes: máximo, mínimo, media aritmética, mediana, OWA1 ($a = 0, b = 0.5$), OWA2 ($a = 0.5, b = 1$), OWA3 ($a = 0.3, b = 0.8$), integral Choquet e integral Sugeno.

Dado que vamos a elegir tres funciones de agregación G_1, G_2, G_3 de entre las nueve mencionadas anteriormente, es necesario realizar 9^3 pruebas. Para ello vamos a hacer las 729 pruebas con 30 imágenes, las cuales se dividen en 10 imágenes originales y 20 con ruido digital (variación aleatoria del brillo o el color en las imágenes digitales producido por el dispositivo de entrada) de las cuales 10 con ruido de sal y pimienta (el píxel ruidoso en cuestión no tiene relación alguna con los píxeles circundantes) y 10 con ruido gaussiano (todos y cada uno de los píxeles que componen la imagen cambian su valor, de acuerdo con una distribución normal o gaussiana). La diferencia entre una imagen con o sin ruido puede verse en la Figura 4.2.

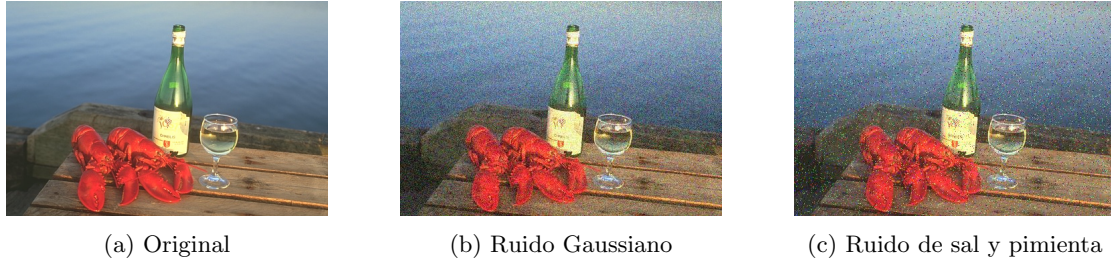


Figura 4.2: Imagen con diferentes alteraciones

La idea será aplicar el algoritmo de reducción de la Sección 3.3 a las imágenes, de las 729 formas, considerando ventanas de diferentes tamaños como, por ejemplo, bloques de píxeles de 2×2 , 4×4 y 8×8 . Posteriormente, usamos el método de magnificación presente en [2] y procedemos a comparar la imagen original con la magnificación con los métodos descritos en la Subsección 2.2.2 del Capítulo 2.

Después de realizar las diferentes pruebas, los mejores resultados los obtenemos usando las agregaciones media aritmética, mediana y OWA1 en este orden.

Una vez elegidas la matriz A_M y las funciones de agregación media aritmética, mediana y OWA1 del algoritmo de reducción, procedemos a aplicar el método de reducción a imágenes de radiografías de pulmón de pacientes con COVID-19. Éstas radiografías tienen múltiples tamaños que van desde los 1500×1500 píxeles hasta los 4200×3500 píxeles. Nosotros las reducimos a dos diferentes tamaños: 32×32 y 128×128 .

Además reduciremos las radiografías usando dos de los mejores métodos comunes de reducción, las agregaciones media aritmética y mediana, para comparar los resultados con los obtenidos con el algoritmo.

En la Figura 4.3 podemos observar la imagen original de una radiografía (Figura 4.3 (a)) con sus diferentes reducciones a los diferentes tamaños mencionados anteriormente. Las Figuras 4.3 (b), (e), (h) se corresponden a la reducción con la agregación media aritmética, las Figuras 4.3 (c), (f), (i) a la reducción con la agregación mediana y las Figuras 4.3 (d), (g), (j) a la reducción con nuestro algoritmo.

Además en las Tabla 4.1 y 4.2 se presentan las comparaciones de las imágenes con los métodos PSNR y el ISI, respectivamente, entre los procedimientos de reducción con media aritmética, mediana y el nuestro algoritmo de reducción. A la vista de los resultados experimentales, que muestran diferencias por debajo de un 1% entre los tres métodos de reducción antes mencionados, no podemos afirmar que mejora a los métodos comunes pero podemos concluir que es un buen método de reducción de imagen.

Estas imágenes reducidas con el algoritmo desarrollado en el Capítulo 3 van a ser utilizadas en el proyecto **SARS2DETECT**, el cual detallaremos en la siguiente sección.



(a) Imagen original



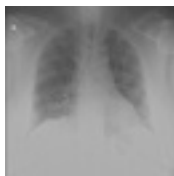
(b)



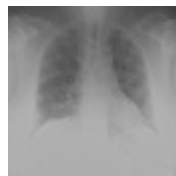
(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)

Figura 4.3: Reducción de imagen a diferentes tamaños con diferentes métodos

Tamaño	PSNR m.aritmetica	PSNR mediana	PSNR algoritmo
32 x 32	38.99	39.28	39.53
64 x 64	41.33	41.57	41.87
128 x 128	41.87	41.93	42.19

Tabla 4.1: Comparación PSNR

Tamaño	ISI m.aritmetica	ISI mediana	ISI algoritmo
32 x 32	0.9803	0.9814	0.9822
64 x 64	0.9835	0.9836	0.98411
128 x 128	0.9763	0.9757	0.9762

Tabla 4.2: Comparación ISI

4.1. Aplicación del algoritmo: Proyecto SARS2DETECT

El objetivo principal de este proyecto es el desarrollo de una herramienta de diagnóstico evolutivo para el daño inducido por la COVID-19, que acompañe a los médicos de primera línea en la toma de decisiones prediciendo cuándo un paciente ya diagnosticado puede evolucionar favorablemente o no.

Por el momento, no están claras las características clínicas específicas de la neumonía por COVID-19 en las diferentes etapas, y no ha sido posible identificar patrones claros que permitan determinar cómo evolucionará la enfermedad ya que existen casos de pacientes que acudieron a urgencias en estado crítico con radiografías casi normales y casos de pacientes con buen estado general que mostraron radiografías alarmantes.

Un modelo capaz de predecir la evolución de un paciente identificaría a los pacientes de alto riesgo, permitiendo el uso de una terapia de soporte respiratorio adecuada lo antes posible, mejorando su pronóstico. Por otro lado, la velocidad de respuesta en milisegundos de un modelo predictivo podría mejorar la eficiencia en primera línea en términos de clasificar la gravedad y la predicción de la tendencia de desarrollo fatal, facilitando el trabajo a los facultativos.

Este proyecto todavía en desarrollo es liderado por NAITEC. También como socios están el Gobierno de Navarra (Servicio de Avance Digital), Complejo Hospitalario Navarro y UPNA.

Entre las pruebas preliminares llevadas a cabo está la de construir un clasificador que pudiera predecir, a partir de las radiografías, la edad, el sexo y su índice de masa corporal (IMC), si el paciente necesita hospitalización o no, UCI, así como su exitus. En esta prueba utilizamos un clasificador Random Forest [6].

Las radiografías son pre-procesadas antes de aplicar técnicas de aprendizaje automático. El pre-procesamiento consiste en la extracción de características de las imágenes, las cuales proporcionan un método de representación de imágenes más adecuado para modelos informáticos, para ello nos basamos en lo desarrollado en [4]. Dichas características se pasan directamente al clasificador. El tamaño de las radiografías, así como del conjunto de entreno pueden exigir mucho tiempo de cálculo. Por esta razón, con el objetivo de reducir dichos tiempos, se ha aplicado el método presentado en este trabajo para reducir el tamaño de las radiografías antes de pasarlas al clasificador.

Esta prueba ha sido realizada con 800 radiografías balanceadas, donde de las 1024 características obtenidas en el pre-procesamiento quitamos las correlacionadas y añadimos las antes mencionadas (edad, sexo, IMC).

Al aplicar clasificador a las imágenes con diferentes tamaños obtenemos lo siguiente:

- Utilizando las imágenes de tamaño 32×32 píxeles, de las 1024 características quitando las correlacionadas nos quedamos con 26. Obtenemos un 100 % de precisión en el entrenamiento y un 51.03 % de precisión en el test. Además en la Figura 4.4 (a) podemos observar la matriz de confusión del test
- Utilizando las imágenes de tamaño 128×128 píxeles, de las 1024 características quitando las correlacionadas nos quedamos con 32. Obtenemos un 100 % de precisión en el entrenamiento y un 52.06 % de precisión en el test. Además en la Figura 4.4 (b) podemos observar la matriz de confusión del test
- Utilizando las imágenes originales, de las 1024 características quitando las correlacionadas nos quedamos con 31. Obtenemos un 100 % de precisión en el entrenamiento y un 57.22 % de precisión en el test. Además en la Figura 4.4 (c) podemos observar la matriz de confusión del test

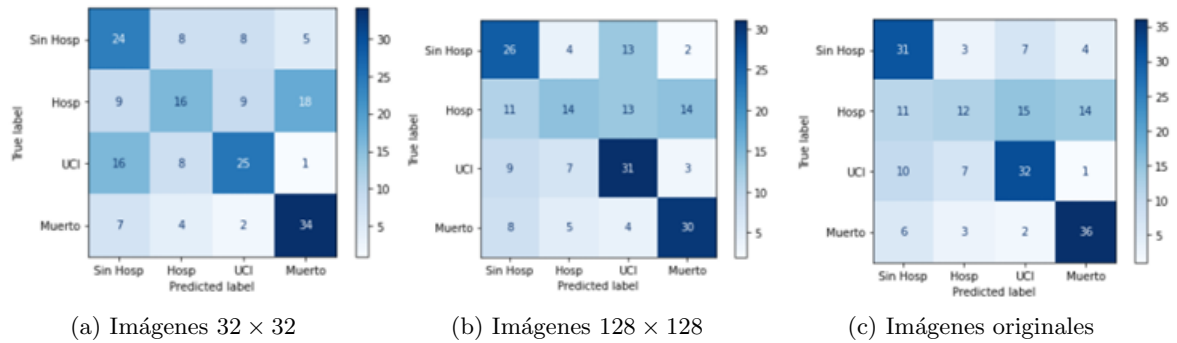


Figura 4.4: Matriz de confusión del test con diferentes tamaños de imagen

Para cuantificar esa reducción en tiempo, con imágenes de 32×32 píxeles el test tenía una velocidad de 30 imágenes por segundo, mientras que con las imágenes originales era de 2.5 imágenes por segundo. Como podemos ver en los resultados obtenidos por los diferentes tests y en la Figura 4.4, observamos que como era de esperar las imágenes de tamaño original devuelven mejores resultados. Sin embargo los obtenidos con las imágenes reducidas no difieren mucho de estos. Por lo tanto en el caso de que hubiera un conjunto de datos muy grande podría ser interesante el uso de esa reducción de tamaño con el objetivo de bajar los tiempos de cálculo.

Capítulo 5

Conclusiones

Durante la realización de este trabajo, he podido percibir la adaptación de una herramienta puramente matemática en algo tangible como es un algoritmo de reducción de imagen. Es más, en este trabajo he podido experimentar un aplicación directa del algoritmo a una situación actual: El diagnóstico evolutivo de pacientes con COVID-19.

En relación al método desarrollado en el Capítulo 4, a la vista de los resultados experimentales podemos afirmar que algoritmo de reducción de imagen tal vez no mejora a los métodos existentes en la literatura, pero sí que supone una herramienta adicional para el tratamiento de las imágenes con un grado de fiabilidad aceptable.

Por otro lado, resulta prometedor el uso que se le puede dar a este algoritmo en una situación tan actual como es la infección por COVID-19, así como la ayuda que proporciona el mismo contribuyendo al diagnóstico evolutivo de los pacientes afectados por este virus.

Como futuras líneas de investigación podríamos preparar el algoritmo desarrollado en el Capítulo 3 para su uso dentro de una red neuronal como una capa de pooling junto con su desarrollo en dimensiones superiores. Además fuera del algoritmo también mi propósito es aprender más sobre redes neuronales para aplicarlas a otros proyectos semejantes a este.

Bibliografía

- [1] A.Castillo-López, Z.Takáč, P.A.Lizarraga, J.Fernandez, and H.Bustince. Aggregation functions on l^1 , l^2 and l^3 with respect to admissible linear orders. *Preprint*, 2021.
- [2] A.Jurio, M.Pagola, R.Mesiar, G.Beliakov, and H.Bustince. Image magnification using interval information. *IEEE transactions on image processing, Vol.20, N^o.11*, November 2011.
- [3] A.Muñoz, T.Blu, and M.Unser. Least-squares image resizing using finite differences. *IEEE Transactions on Image Processing*, pages 1365–1378, September 2001.
- [4] Joseph Paul Cohen. Chester the ai radiology assistant. <https://github.com/mlmed/chester-xray>, February 2021.
- [5] D.Paternain, J.Fernandez, H.Bustince, R.Mesiar, and G.Beliakov. Construction of image reduction operators using averaging aggregation functions. *Fuzzy Sets and Systems 261*, pages 87–111, 2015.
- [6] Jose Martinez Heras. Random forest (bosque aleatorio): combinando árboles. <https://www.iartificial.net/random-forest-bosque-aleatorio/>, February 2021.
- [7] I.Perfilieva. Fuzzy transforms:theory and applications. *Fuzzy Sets Syst.157*, pages 993–1023, 2006.
- [8] Suman K.Mitra, C.A.Murthy, and Malay K.Kundu. A technique for image magnification using partitioned iterative function system. *Pattern Recognition 33*, pages 1119–1133, 2000.
- [9] F. Di Martino and S.Sessa. Compression and decompression of images with discrete fuzzy transform. *Inf. Sci. 177*, pages 2349–2362, 2007.
- [10] L. De Miguel, M.Sesma-Sara, M.Elcano, M.Asian, and H.Bustince. An algorithm for group decision making using n-dimensional fuzzy sets, admissible orders and owa operators. *Information Fusion 37*, pages 126–131, 2017.
- [11] Raghav Prabhu. 10 cnn architectures. <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d>, February 2021.
- [12] Raghav Prabhu. Understanding of convolutional neural network(cnn)-deep learning. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>, February 2021.
- [13] Andreas Refsgaard, Francis Tseng, and Gene Kogan. How neural networks are trained. https://ml4a.github.io/ml4a/how_neural_networks_are_trained/, February 2021.