

**DOBLE GRADO EN
INGENIERÍA INFORMÁTICA Y MATEMÁTICAS**



**UNIVERSIDAD
DE GRANADA**

TRABAJO FIN DE GRADO

**Desarrollo de algoritmos para la
aproximación no lineal mediante bases
especiales en espacios de Banach**

Carlos Santiago Sánchez Muñoz

Tutores

Francisco Javier Merí de la Maza
Alberto Fernández Hilario

15 de junio de 2020



Facultad de Ciencias



E.T.S. Ingenierías
Informática y de Telecomunicación

Desarrollo de algoritmos para la aproximación no lineal mediante bases especiales en espacios de Banach

Carlos Santiago Sánchez Muñoz

Palabras clave: bases greedy, aproximación no lineal, sistema de Haar, aproximación de señales y compresión de imágenes.

Resumen

El Análisis Funcional es una rama moderna del Análisis Matemático en donde se considera el enfoque de ver las funciones como elementos o puntos de un espacio vectorial más abstracto con unas ciertas propiedades denominado espacio de Banach. Este trabajo se enmarca en una subrama conocida como Geometría de los Espacios de Banach con el objeto de estudiar las bases y sucesiones básicas de tales espacios para comprender las similitudes y diferencias de los mismos. A parte de esta teoría isomórfica, en los años 90 surge el concepto de base greedy que está relacionado con la aproximación no lineal y la construcción de algoritmos de compresión de información.

La aproximación de una función por una combinación lineal de vectores elegidos de una colección prefijada tiene una gran variedad de aplicaciones. El problema estándar de aproximación greedy es el de la aproximación de m términos donde uno fija una base y busca aproximar una función objetivo mediante una combinación lineal de m términos de la base. Estamos interesados en los aspectos cuantitativos de esta forma de aproximación. Un problema de optimización de este tipo es muy difícil y es útil tener algoritmos de aproximación eficientes en tiempo para resolver problemas atractivos de compresión.

Los objetivos que se persiguen en este trabajo son familiarizarse con conceptos y propiedades de bases así como de tipos de bases y ejemplos de dichos tipos con el propósito de examinar los posibles vínculos y aportaciones a la aproximación no lineal. Por otro lado, se pretende presentar modelos teóricos de aproximación basados en este desarrollo y elegir ejemplos prácticos donde trasladar los modelos teóricos. Asimismo es necesario utilizar técnicas de aprendizaje para optimizar los parámetros del problema buscando la eficiencia en precisión y tiempo.

Los resultados han sido excelentes. La clave para realizar aproximaciones implementables y eficientes usando bases greedy es usar la transformada de Haar discreta haciendo que éste sea el punto de partida para los algoritmos de compresión. Los ejemplos de compresión seleccionados son la compresión de imágenes y el procesamiento de señales. En ellos se obtienen resultados reveladores en donde la optimización de ciertos parámetros

resulta necesaria de cara a obtener compresiones muy buenas.

Development of algorithms for non-linear approximation through special bases in Banach spaces

Carlos Santiago Sánchez Muñoz

Keywords: greedy bases, non-linear approximation, Haar system, signal approximation and image compression.

Abstract

Functional Analysis is a modern branch of Mathematical Analysis where the approach of seeing functions as elements or points of a more abstract vector space with certain properties called Banach space is considered. This work is framed in a sub-branch known as Geometry of Banach Spaces in order to study the bases and basic sequences of such spaces to understand their similarities and differences. Apart from this isomorphic theory, in the 90's the concept of greedy base arises, which is related to the nonlinear approximation and the construction of information compression algorithms.

The approximation of a function by a linear combination of vectors chosen from a prefixed collection has a wide variety of applications. The standard greedy approximation problem is the m -term approximation where one sets a basis and seeks to approximate an objective function by a linear combination of m terms of the basis. We are interested in the quantitative aspects of this form of approximation. Such an optimization problem is very difficult and it is useful to have time efficient approximation algorithms to solve attractive compression problems.

The objectives pursued in this work are to become familiar with concepts and properties of basis as well as types of basis and examples of these types in order to examine possible links and contributions to the nonlinear approximation. On the other hand, the aim is to present theoretical approximation models based on this development and choose practical examples where to transfer the theoretical models. Likewise, it is necessary to use learning techniques to optimize the parameters of the problem, seeking efficiency in precision and time.

The results have been excellent. The key to making efficient and implementable approaches using greedy bases is to use the discrete Haar transform making this the starting point for compression algorithms. Selected examples of compression are image compression and signal processing. Revealing results are obtained where optimization of certain parameters is necessary in order to obtain very good compressions.

The contents of this Final Undergraduate Thesis are of two kinds: mathematical and

computational. The most theoretical and **mathematical** part is divided into four sections. Most of the results have been collected from the Albiac and Kalton monograph [3].

Bases and basic sequences. Let us focus first on the work environment: we define a basis $(e_n)_{n=1}^{\infty}$ for an arbitrary Banach space X . We continue defining the biorthogonal functionals $(e_n^*)_{n=1}^{\infty}$ and the partial sums projections $(S_n)_{n=0}^{\infty}$ associated with that basis. We introduce the basic constant K_b of $(e_n)_{n=1}^{\infty}$ and in the case that $K_b = 1$ the basis is said to be monotone.

Secondly, we analyse some examples of Fourier series and prove that the spaces $\mathcal{C}[0, 1]$ and $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$ have a monotone basis and that the trigonometric system fails to be a basis of $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$. This section concludes with the definition of another relevant concept in this paper, basic sequences and its equivalents and characterizations.

Special type of bases. In this section we study four special types of bases and some of their properties. Firstly, the unconditional bases in which the series of an element $x \in X$ unconditionally converges to it. Then we examine the symmetric bases as a particular case of the unconditional ones where the basis $(e_n)_{n=1}^{\infty}$ is equivalent to $(e_{\pi(n)})_{n=1}^{\infty}$ for all permutation π of \mathbb{N} . Finally, the notions of boundedly complete bases and shrinking bases are introduced.

The spaces L_p for $1 \leq p < \infty$. We begin the section introducing the Haar system that will be fundamental in the central result of this work and it will be an indispensable tool in the computational part. We will continue proving that the Haar system is a monotone basis for $L_p[0, 1]$ with $1 \leq p < \infty$. Finally, we will deduce that the Haar system is an unconditional basis and calculate its best unconditional constant.

Greedy-type bases. This is the main section of the work. First, we define the m -term approximation of an element x and the error we obtain in the approximation. If \mathcal{B} is a basis of X then for each $m \in \mathbb{N}$ we let $\Sigma_m[\mathcal{B}, X]$ denote the collection of all the elements of X that can be expressed as a linear combination of m elements of \mathcal{B} . The fundamental question is how to construct for each $x \in X$ and $m \in \mathbb{N}$ an element $y_m \in \Sigma_m[\mathcal{B}, X]$ such that $\|x - y_m\|$ is small. If X is a Hilbert space and \mathcal{B} is an orthonormal basis then we look for the elements of the basis for which $|\langle x, e_n \rangle|$ is largest. Afterwards we get some properties of $\mathcal{G}_m(x)$, the greedy algorithm of x , which is uniquely determined.

We continue with the definition of democratic bases as the last type of bases that we are going to study. A basis \mathcal{B} is said to be democratic if blocks of the same size of \mathcal{B} have uniformly comparable norms. The democracy functions and two useful lemmas will be used to achieve the final result of this study.

In order to finish the mathematical part, we define the concept of greedy basis: a basis \mathcal{B} of a Banach space X is greedy if there is an absolute constant C such that

$\|x - \mathcal{G}_m[\mathcal{B}, X](x)\| \leq C \inf_{y \in \Sigma_m[\mathcal{B}, X]} \|x - y\|$ for all $m \in \mathbb{N}$ and $x \in X$. Next we enunciate a very satisfactory characterization given by Konyagin and Temlyakov that says that a basis is greedy if and only if it is unconditional and democratic. Finally, by this theorem proving that the Haar system is democratic, we conclude that it is also a greedy basis in $L_p[0, 1]$ ($1 < p < \infty$) arriving at the central result that we desire.

The possibility of replacing an object by a simpler one which provides the same information for practical purposes is very useful. We begin with the development of two theoretical algorithms to make approximations using as greedy basis a dictionary \mathcal{D} . Each algorithm will focus on the environment of a different space: (1) a Hilbert space and (2) an arbitrary Banach space X .

The practical and **computer-based** part is divided into three parts.

From theoretical algorithms to implementation. Discretizing the normalized Haar system we obtain a very satisfactory procedure for information compression. In this section this process is designed, from the achievement of the coefficients to the reconstruction through the run-length encoding. Furthermore, the design includes the option to normalize in any $L_p[0, 1]$ space.

Haar wavelet for signal compression. This section is the first application of the built model. Technical aspects of signal discretization and subsequent compression are explained. Later tests are carried out on two signals and the results are analyzed. The algorithm parameters are also optimized.

Haar wavelet for image compression. For this application it is essential to extend the transform model to the bidimensional case. Next, the results are analyzed in a similar way to what was done in the signals. Finally, a geometric interpretation of the image of the coefficients is given and the optimization of the algorithm parameters is investigated.

Yo, **Carlos Santiago Sánchez Muñoz**, alumno de la titulación DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación y Facultad de Ciencias de la Universidad de Granada**, con DNI 75931715K, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.



Fdo: Carlos Santiago Sánchez Muñoz

Granada a 15 de junio de 2020.

D. **Francisco Javier Merí de la Maza**, Profesor del Departamento Análisis Matemático de la Universidad de Granada.

D. **Alberto Fernández Hilario**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Desarrollo de algoritmos para la aproximación no lineal mediante bases especiales en espacios de Banach*, ha sido realizado bajo su supervisión por **Carlos Santiago Sánchez Muñoz**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 15 de junio de 2020.

Los directores:

Francisco Javier Merí de la Maza

Alberto Fernández Hilario

Agradecimientos

En primer lugar quiero agradecer a mis padres y a mis hermanos su cariño, incondicionalidad y constante apoyo hacia mí. Ellos son lo más importante para cualquier meta que me propongo.

Quiero agradecer también a todos los profesores con los que he tenido contacto en estos cinco años, han realizado una gran labor docente y de algunos he aprendido mucho más que matemáticas o informática.

Del mismo modo he sentido la fuerza que me han dado todos mis amigos, algunos de los cuales los conocí en esta carrera y junto a ellos he superado obstáculos y dificultades.

Por último quiero darle las gracias de una manera especial a mis tutores, Javier Merí y Alberto Fernández, quienes me han dado las herramientas para llevar a cabo este Trabajo Fin de Grado. A Javier por todo lo que he podido compartir con él, por su constante trabajo y apoyo y porque la idea de hacer este bonito trabajo es suya. A Alberto por ser un gran experto en su área de conocimiento, por tener siempre la respuesta a mis dudas y por su apoyo.

Índice general

Resumen y palabras clave	I
Abstract and keywords	III
Autorización de ubicación del trabajo en la biblioteca	VII
Autorización para la defensa del Trabajo Fin de Grado	IX
Agradecimientos	XI
1. Introducción	3
1.1. Contexto	3
1.2. Justificación y propuesta	4
1.3. Contenidos	5
1.4. Materias relacionadas con el trabajo	6
1.5. Objetivos	7
1.5.1. Estudio del estado del arte	7
1.5.2. Diseño e implementación de algoritmos de compresión	7
2. Desarrollo matemático del trabajo	9
2.1. Bases y sucesiones básicas	9
2.1.1. Bases de Schauder	9
2.1.2. Ejemplos: Series de Fourier	14
2.1.3. Equivalencia de bases y sucesiones básicas	21
2.2. Tipos especiales de bases	24
2.2.1. Bases incondicionales	24
2.2.2. Bases simétricas	31
2.2.3. Bases y dualidad: bases acotadamente completas y bases shrinking .	33
2.3. La base de Haar en $L_p[0, 1]$ ($1 \leq p < \infty$)	42
2.4. Bases de tipo greedy	49
2.4.1. Marco general	50

2.4.2. Bases democráticas	52
2.4.3. Bases greedy	54
2.4.4. Algoritmos greedy teóricos	58
3. Desarrollo informático del trabajo	61
3.1. De los algoritmos teóricos a la implementación	61
3.1.1. Marco	62
3.1.2. Transformada de Haar discreta	63
3.1.3. Normalización de la transformada en $L_p[0, 1]$	65
3.1.4. Implementación de la transformada de Haar discreta	67
3.1.5. Compresión de los coeficientes	71
3.2. Transformada de Haar para aproximación de señales	72
3.2.1. Diseño de la solución	73
3.2.2. Algoritmos	74
3.2.3. Resultados y análisis	75
3.2.4. Optimización del error medio y la compresión	79
3.3. Transformada de Haar para compresión de imágenes	84
3.3.1. Diseño de la solución	85
3.3.2. Algoritmos	87
3.3.3. Resultados y análisis	91
3.3.4. Optimización del error medio y la compresión	103
4. Conclusiones y Trabajos Futuros	107
4.1. Conclusiones	107
4.2. Trabajos futuros	109
Bibliografía	109

Capítulo 1

Introducción

En este capítulo se presentan los contenidos que se van a tratar a lo largo del trabajo. Para ello exponemos el contexto en el que se enmarca y la justificación del interés en realizar este estudio para lo cual se plantean diversos problemas. Por último, se concretan los objetivos a seguir.

1.1. Contexto

Este Trabajo Fin de Grado se enmarca dentro de la rama del Análisis Funcional conocida como Geometría de los Espacios de Banach [1, 2], y tiene por objeto avanzar en el estudio de bases especiales en tales espacios. Una de las ideas principales en el estudio isomórfico de espacios de Banach es el uso de las propiedades de bases y sucesiones básicas como herramienta fundamental para comprender las diferencias y similitudes entre distintos espacios [3].

En los años 90 del siglo pasado surge el concepto de base greedy que está íntimamente relacionado con la aproximación no lineal. Este hecho supuso el comienzo de la búsqueda de algoritmos que se pueden implementar en computadoras para realizar compresión de datos y recuperación de información [10]. Comienza así la construcción de modelos teóricos de aproximación basados en la teoría desarrollada a partir de ese concepto con el objetivo de realizar compresión de información en diversos ejemplos [11].

En la actualidad los volúmenes de datos que se manejan son cada vez más grandes. Por este motivo la compresión de los mismos tiene gran utilidad ya que ayuda tanto a mejorar su almacenaje como su transmisión. Este hecho le da una gran relevancia a la propuesta de este trabajo. Además, analizar un modelo de compresión basado en la herramienta de bases en espacios de Banach es cuanto menos novedoso.

1.2. Justificación y propuesta

Hay una serie de situaciones en las que es muy útil o incluso imprescindible realizar compresión de información para que el objeto a manejar sea más simple pero que aporte prácticamente la misma información. Por ejemplo, si queremos enviar una señal a otro planeta puede que necesitemos comprimir lo máximo posible dicha información.

Esta aplicación del trabajo es muy relevante y es posible gracias a la aproximación y compresión de datos. Aquí reside el interés de este proyecto. El primer paso es desarrollar el aspecto matemático y familiarizarse con las bases y sus propiedades sabiendo que la meta es la aproximación no lineal [3]. A continuación se centra la atención en las bases greedy que son buenas aproximadoras y se estudia la contribución que pueden hacer [9]. Del mismo modo surge la necesidad de examinar ejemplos concretos de esas bases para ver sus aportaciones y las posibilidades de trasladarlas a un computador [10].

En un segundo paso se analizan varios algoritmos teóricos de aproximación mediante las bases greedy [11] que serán fáciles de entender gracias al trabajo matemático pero poco implementables. Ante esta dificultad aparece el objetivo de buscar modelos implementables basados en esta teoría. El hecho de que el sistema de Haar sea greedy en $L_p[0, 1]$ con $1 < p < \infty$ es la llave para realizar la aproximación.

En este trabajo se utilizan herramientas matemáticas muy abstractas para abordar computacionalmente problemas concretos de compresión. Esto supone una conexión efectiva de dos mundos que viven separados para los especialistas en uno sólo de los campos.

Una vez conocido el estado del arte, la propuesta que se hace es usar la transformada de Haar discreta, la cual es implementable y puede ofrecer gran beneficio en bastantes ejemplos discretos [12, 13]. Hay grandes contribuciones acerca de transformadas wavelet pero esta investigación se va a centrar en la de Haar y en trasladar la teoría que hay a ejemplos que más adelante se proponen a la vez que interpretar los hallazgos y optimizar mediante técnicas de aprendizaje los diferentes parámetros del problema.

Los ejemplos propuestos para trabajar estos algoritmos son dos:

- Aproximación de señales. Las señales son indispensables en la actualidad. Estamos continuamente enviando y recibiendo señales a diferentes puntos. Aquí se desea adaptar los modelos presentados anteriormente para la compresión de las mismas [14].
- Compresión de imágenes. Los ordenadores emplean gráficos ampliamente y las imágenes están presentes en muchas áreas de la vida cotidiana. De este modo, las imágenes son importantes, ¡pero tienden a ser grandes! El hardware moderno puede mostrar muchos colores, por lo que es común tener un píxel representado internamente como

un número de 24 bits, donde los porcentajes de rojo, verde y azul ocupan 8 bits cada uno. Tal píxel de 24 bits puede especificar uno de $2^{24} \approx 24,78$ millones de colores. Como resultado, una imagen con una resolución más bien pequeña como 512×512 que conste de tales píxeles ocupa 786.432 bytes. Es por eso que la compresión de imágenes es tan importante.

Una característica importante de la compresión de imágenes es que puede tener pérdidas. Muchas veces resulta aceptable perder características de imagen a las que el ojo no es sensible. Ésta es una de las ideas principales detrás de los muchos métodos de compresión de imágenes. Aquí se pretende implementar un algoritmo de compresión basado en la transformada de Haar discreta optimizando diferentes parámetros del problema.

1.3. Contenidos

La parte más teórica y **matemática** está agrupada en cuatro secciones.

Bases y sucesiones básicas. Ubica el ambiente de trabajo: se define el concepto de base $(e_n)_{n=1}^{\infty}$ para un espacio de Banach X . De la misma forma se definen los funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$ y las proyecciones $(S_n)_{n=0}^{\infty}$ asociadas a la base así como el concepto de sucesión básica. En segundo lugar se analizan ejemplos de series de Fourier y se prueba que el sistema trigonométrico no es una base de $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$.

Tipos especiales de bases. En este apartado se introducen cuatro tipos especiales de bases y algunas propiedades de estas: bases incondicionales, bases simétricas como un caso particular de las incondicionales, bases acotadamente completas y bases shrinking.

Los espacios L_p para $1 \leq p < \infty$. Se presenta el sistema de Haar que más adelante será una herramienta indispensable en la parte informática. Se demuestra que es una base monótona e incondicional para $L_p[0, 1]$ con $1 < p < \infty$.

Bases de tipo greedy. Este es el apartado objetivo. En primer lugar se denota $\Sigma_m[\mathcal{B}, X]$ como la colección de todos los elementos de X que pueden ser expresados mediante una combinación lineal de m elementos de la base \mathcal{B} . La pregunta fundamental es cómo construir para cada $x \in X$ y $m \in \mathbb{N}$ un elemento $y_m \in \Sigma_m[\mathcal{B}, X]$ de manera que $\|x - y_m\|$ sea pequeño. Después de esto se define el concepto de base democrática. Finalmente, se expone el concepto de base greedy y uno de los teoremas más importantes: la caracterización de Konyagin y Temlyakov. Ésta es la clave para probar que el sistema de Haar es una base greedy en $L_p[0, 1]$ ($1 < p < \infty$) llegando al resultado central deseado.

Se presentan dos algoritmos teóricos para realizar aproximaciones usando como base greedy un diccionario \mathcal{D} . Los dos algoritmos dependen del ambiente de trabajo: (1) un

espacio de Hilbert y (2) un espacio de Banach arbitrario.

La parte práctica y de un carácter **informático** está agrupada en tres secciones.

De los algoritmos teóricos a la implementación. Discretizando el sistema de Haar normalizado se obtiene un procedimiento satisfactorio para compresión de información. En esta sección se diseña este proceso, desde la consecución de los coeficientes hasta la reconstrucción pasando por la codificación de longitud. Además el diseño incluye la opción de normalizar en cualquier espacio $L_p[0, 1]$.

Transformada de Haar para aproximación de señales. Esta sección es la primera aplicación del modelo construido. Se explican aspectos técnicos de la discretización de la señal y la posterior compresión. Más adelante se realizan pruebas sobre dos señales y se analizan los resultados. Asimismo se optimizan los parámetros del algoritmo.

Transformada de Haar para compresión de imágenes. Para esta aplicación es imprescindible extender el modelo de la transformada al caso bidimensional. A continuación se analizan los resultados de manera análoga a como se hizo en las señales. Por último, se da una interpretación geométrica de la imagen de los coeficientes y se investiga la optimización de los parámetros del algoritmo.

1.4. Materias relacionadas con el trabajo

Se van a enumerar aquellas asignaturas más relacionadas con el trabajo. Éstas son un prerequisito para su correcto desarrollo.

Las materias del Grado de Matemáticas son:

- Geometría I.
- Cálculo I y II.
- Análisis Matemático I y II.
- Variable Compleja I.
- Análisis Funcional.

Las del Grado en Ingeniería Informática son:

- Fundamentos de Programación.
- Metodología de la Programación.
- Estructuras de Datos.
- Algorítmica.
- Visión por Computador.

1.5. Objetivos

Los objetivos marcados son de dos índoles: matemáticos e informáticos. Los primero estará dirigidos a conocer el estado del arte del problema; los segundos al diseño de aplicaciones de las bases a la compresión de información.

1.5.1. Estudio del estado del arte

Los objetivos matemáticos del trabajo están orientados a familiarizarse con las bases en el ambiente de un espacio de Banach y comprender las propiedades de éstas para estudiar las aportaciones que pueden hacer a la aproximación no lineal.

- 1) Estudiar el concepto de base en espacios de Banach, presentando ejemplos y analizando las distintas propiedades que pueden tener las bases.
- 2) Presentar tipos especiales de bases en estos espacios. Asimismo estudiar caracterizaciones y propiedades relevantes de las bases mencionadas anteriormente.
- 3) Presentar el sistema de Haar y probar que es una base incondicional para los espacios $L_p[0, 1]$ donde $1 < p < \infty$.
- 4) Por último, pero no menos importante, estudiar las bases greedy y la relación que éstas tienen con la aproximación no lineal. Examinar la caracterización de Konyagin y Temlyakov para probar que el sistema de Haar es greedy en $L_p[0, 1]$ ($1 < p < \infty$).

Los objetivos matemáticos se han alcanzado durante el desarrollo del trabajo y el grado de consecución de éstos es muy alto.

1.5.2. Diseño e implementación de algoritmos de compresión

Los objetivos informáticos marcados en este proyecto son:

- 1) Presentar y analizar modelos teóricos de aproximación mediante bases greedy.
- 2) Buscar ejemplos prácticos sobre los que implementar y testear los modelos.
- 3) Utilizar técnicas de aprendizaje y / o optimización para determinar los parámetros de las funciones bajo estudio sobre los modelos elegidos trabajando de manera automática por los grados de libertad que tiene el problema y con la necesidad de la eficiencia en la obtención de la solución, equilibrando precisión y tiempo de cómputo.

En el desarrollo del trabajo se ha elegido como ejemplo de compresión de información la compresión de señales e imágenes. El estudio realizado y el guión seguido es similar en los dos casos.

Los objetivos informáticos que nos hemos marcado se han conseguido con un grado de

satisfacción muy alto.

Capítulo 2

Desarrollo matemático del trabajo

El aspecto matemático del trabajo consiste en un estudio de bases y sucesiones básicas en espacios de Banach y sus aportaciones a la teoría de aproximación.

En primer lugar se estudia la definición de base y algunas propiedades y ejemplos. A continuación presentaremos tipos especiales de bases así como las caracterizaciones y relaciones entre ellas. Más adelante nos centraremos en el caso $L_p[0, 1]$ ($1 \leq p < \infty$) y presentaremos el sistema de Haar como base de ese espacio. Por último, se tratará el concepto de base greedy con algunos ejemplos y resultados importantes.

2.1. Bases y sucesiones básicas

Una de las ideas clave en la teoría isomórfica de espacios de Banach es usar las propiedades de las bases y sucesiones básicas para estudiar las diferencias y similitudes entre espacios. Durante el desarrollo convendremos en que todos los espacios de Banach son reales a menos que se indique lo contrario y en algunas ocasiones probaremos el caso complejo para demostrar que se obtiene el mismo resultado.

2.1.1. Bases de Schauder

Aunque la noción de base de un espacio vectorial tiene sentido para espacios de dimensión infinita, no resulta satisfactoria en el ambiente de los espacios normados porque no tiene en cuenta su estructura topológica. Pasamos a definir el concepto adecuado de base en este ambiente.

2.1.1 Definición. Una sucesión de elementos $(e_n)_{n=1}^{\infty}$ en un espacio de Banach de dimensión infinita se dice que es una base de X si para cada $x \in X$ existe una única sucesión de escalares $(a_n)_{n=1}^{\infty}$ tal que

$$x = \sum_{n=1}^{\infty} a_n e_n.$$

Esto significa que necesitamos que $(\sum_{n=1}^N a_n e_n)_{N=1}^{\infty}$ converja a x en la topología de la norma.

De la definición es claro que una base consiste en vectores linealmente independientes y distintos de cero. Si X tiene una base $(e_n)_{n=1}^{\infty}$ entonces su cierre coincide con X y por tanto X es separable.

Nos damos también cuenta de que si $(e_n)_{n=1}^{\infty}$ es una base de un espacio de Banach X , las aplicaciones $x \mapsto a_n$ son funcionales lineales en X . Escribiremos $e_n^{\#}(x) = a_n$. Sin embargo, no es de ninguna manera inmediato ver que los funcionales lineales $(e_n^{\#})_{n=1}^{\infty}$ son en realidad continuos. Hagamos la siguiente definición:

2.1.2 Definición. Sea $(e_n)_{n=1}^{\infty}$ una sucesión en un espacio de Banach X . Supongamos que hay una sucesión $(e_n^*)_{n=1}^{\infty}$ en X^* tal que

- (I) $e_k^*(e_j) = 1$ si $j = k$ o bien $e_k^*(e_j) = 0$ si $j \neq k$ para todo k y j en \mathbb{N} ,
- (II) $x = \sum_{n=1}^{\infty} e_n^*(x) e_n$ para cada x en X .

Entonces decimos que $(e_n)_{n=1}^{\infty}$ es una base de Schauder de X y los funcionales $(e_n^*)_{n=1}^{\infty}$ son los funcionales biortogonales asociados a $(e_n)_{n=1}^{\infty}$.

Si $(e_n)_{n=1}^{\infty}$ es una base de Schauder de X y $x = \sum_{n=1}^{\infty} e_n^*(x) e_n$, el soporte de x es el subconjunto de enteros n tales que $e_n^*(x) \neq 0$. Lo denotamos como $sop(x)$. Si $|sop(x)| < \infty$ diremos que x tiene soporte finito.

El nombre de Schauder en la definición anterior es en honor a J. Schauder quien introdujo el concepto de base en 1927. En la práctica, cualquier base de un espacio de Banach es una base de Schauder. La prueba de la equivalencia entre los conceptos de base y base de Schauder es consecuencia del teorema de la gráfica cerrada. Aunque la demostración de este resultado es muy interesante porque hace uso de algunos de los principios básicos del análisis funcional, hay que reconocer que es esencialmente inútil en el sentido de que en todas las situaciones prácticas somos capaces de demostrar que $(e_n)_{n=1}^{\infty}$ es una base mostrando la conclusión formalmente más fuerte de que ya es una base de Schauder.

2.1.3 Teorema. *Sea X un espacio de Banach (separable). Una sucesión $(e_n)_{n=1}^{\infty}$ en X es una base de Schauder si y solo si $(e_n)_{n=1}^{\infty}$ es una base para X .*

Demostración. Asumimos que $(e_n)_{n=1}^{\infty}$ es una base de X e introducimos las *proyecciones* $(S_n)_{n=0}^{\infty}$ asociadas a $(e_n)_{n=1}^{\infty}$ definidas por $S_0 = 0$ y para $n \geq 1$,

$$S_n(x) = \sum_{k=1}^n e_k^{\#}(x) e_k.$$

Por supuesto, no sabemos todavía que estos operadores están acotados. Consideraremos una nueva norma en X definida por la fórmula

$$|||x||| = \sup_{n \geq 1} \|S_n x\|.$$

Como $\lim_{n \rightarrow \infty} \|x - S_n x\| = 0$ para cada $x \in X$, se sigue que $|||\cdot||| \geq \|\cdot\|$. Demostraremos que $(X, |||\cdot|||)$ es completo.

Supongamos que $(x_n)_{n=1}^{\infty}$ es una sucesión de Cauchy en $(X, |||\cdot|||)$. Por supuesto, $(x_n)_{n=1}^{\infty}$ converge a algún $x \in X$ en la norma original. Nuestro objetivo es probar que $\lim_{n \rightarrow \infty} |||x_n - x||| = 0$.

Para cada k fijado la sucesión $(S_k x_n)_{n=1}^{\infty}$ es convergente en la norma original a algún $y_k \in X$, nos damos cuenta también de que $(S_k x_n)_{n=1}^{\infty}$ está contenido en el subespacio finito dimensional generado por e_1, \dots, e_k , el cual notamos por $[e_1, \dots, e_k]$. Ciertamente, los funcionales $e_j^{\#}$ son continuos en cada subespacio finito dimensional; por tanto si $1 \leq j \leq k$ tenemos

$$\lim_{n \rightarrow \infty} e_j^{\#}(x_n) = e_j^{\#}(y_k) := a_j.$$

Argumentamos ahora que $\sum_{j=1}^{\infty} a_j e_j = x$ para la norma original.

Dado $\varepsilon > 0$, cogemos un entero n tal que si $m \geq n$ entonces $|||x_m - x_n||| \leq \frac{\varepsilon}{3}$, y tomamos k_0 tal que $k \geq k_0$ implica $\|x_n - S_k x_n\| \leq \frac{\varepsilon}{3}$. Entonces para $k \geq k_0$ tenemos

$$\|y_k - x\| \leq \lim_{m \rightarrow \infty} \|S_k x_m - S_k x_n\| + \|S_k x_n - x_n\| + \lim_{m \rightarrow \infty} \|x_m - x_n\| \leq \varepsilon.$$

Así $\lim_{k \rightarrow \infty} \|y_k - x\| = 0$ y por la unicidad de la expansión de x con respecto a la base obtenemos $S_k x = y_k$.

Ahora,

$$|||x_n - x||| = \sup_{k \geq 1} \|S_k x_n - S_k x\| \leq \limsup_{m \rightarrow \infty} \sup_{k \geq 1} \|S_k x_n - S_k x_m\|,$$

así que el $\lim_{n \rightarrow \infty} |||x_n - x||| = 0$ y $(X, |||\cdot|||)$ es completo.

Por el teorema de la aplicación abierta la aplicación $Id : (X, \|\cdot\|) \rightarrow (X, |||\cdot|||)$ está acotada, i.e., existe K tal que $|||x||| \leq K \|x\|$ para $x \in X$. Esto implica que

$$\|\mathbf{S}_n x\| \leq K \|x\|, \quad x \in X, \quad n \in \mathbb{N}.$$

En particular,

$$|e_n^\#(x)| \|e_n\| = \|\mathbf{S}_n x - \mathbf{S}_{n-1} x\| \leq 2K \|x\|;$$

por lo que $e_n^\# \in X^*$ y $\|e_n^\#\| \leq 2K \|e_n\|^{-1}$. \square

Sea $(e_n)_{n=1}^\infty$ una base de X . El teorema anterior nos dice que $(e_n)_{n=1}^\infty$ es de hecho una base de Schauder; de aquí que usemos la notación $(e_n^*)_{n=1}^\infty$ para los funcionales biortogonales.

Consideramos los operadores $\mathbf{S}_n : X \rightarrow X$ dados por $S_0 = 0$ y para $n \geq 1$

$$\mathbf{S}_n \left(\sum_{k=1}^{\infty} e_k^*(x) e_k \right) = \sum_{k=1}^n e_k^*(x) e_k.$$

\mathbf{S}_n es un operador lineal y continuo ya que e_k^* es continuo. Que los operadores $(\mathbf{S}_n)_{n=1}^\infty$ estén uniformemente acotados es consecuencia del principio de acotación uniforme. Recogemos este resultado:

2.1.4 Proposición. *Sea $(e_n)_{n=1}^\infty$ es una base de Schauder de un espacio de Banach X y $(\mathbf{S}_n)_{n=1}^\infty$ las proyecciones naturales asociadas a él. Entonces:*

$$\sup_n \|\mathbf{S}_n\| < \infty.$$

Demostración. Para una base de Schauder los operadores $(\mathbf{S}_n)_{n=1}^\infty$ están acotados a priori. Como $\mathbf{S}_n(x) \rightarrow x$ para cada $x \in X$ tenemos que $\sup_n \|\mathbf{S}_n(x)\| < \infty$ para cada $x \in X$. Entonces el principio de acotación uniforme nos dice que $\sup_n \|\mathbf{S}_n\| < \infty$. \square

2.1.5 Definición. *Si $(e_n)_{n=1}^\infty$ es una base de un espacio de Banach X entonces el número*

$$K_b = \sup_n \|\mathbf{S}_n\|$$

se llama la constante básica. En el caso óptimo de que $K_b = 1$ la base $(e_n)_{n=1}^\infty$ se dice monótona.

2.1.6 Observación. Siempre podemos renormar un espacio de Banach X con una base de manera que la base dada sea monótona. Simplemente ponemos:

$$|||x||| = \sup_{n \geq 1} \|\mathbf{S}_n x\|.$$

Tenemos $\|x\| \leq |||x||| \leq K_b \|x\|$ por lo que la nueva norma es equivalente a la antigua y rápidamente verificamos que $|||S_n||| = 1$ para cada $n \in \mathbb{N}$.

El siguiente resultado establece un método para construir una base para un espacio de Banach X siempre que tengamos una familia de proyecciones que cumplan las propiedades de los operadores de suma parcial.

2.1.7 Proposición. *Supongamos que $S_n : X \rightarrow X, n \in \mathbb{N}$ es una sucesión de proyecciones ($S_n^2 = S_n$) lineales acotadas en un espacio de Banach X tal que*

- (I) $\dim S_n(X) = n$ para cada n ;
- (II) $S_n S_m = S_m S_n = S_{\min\{n,m\}}$ para cualesquiera números naturales m y n ; y
- (III) $S_n(x) \rightarrow x$ para cada $x \in X$.

Entonces toda sucesión de vectores distintos de cero $(e_k)_{k=1}^\infty$ en X escogidos inductivamente de manera que $e_1 \in S_1(X)$ y $e_k \in S_k(X) \cap S_{k-1}^{-1}(0)$ si $k \geq 2$ es una base para X con proyecciones asociadas $(S_n)_{n=1}^\infty$.

Demostración. Sea $0 \neq e_1 \in S_1(X)$ y definimos $e_1^* : X \rightarrow \mathbb{R}$ por $e_1^*(x)e_1 = S_1(x)$. A continuación cogemos $0 \neq e_2 \in S_2(X) \cap S_1^{-1}(0)$ y definimos el funcional $e_2^* : X \rightarrow \mathbb{R}$ por $e_2^*(x)e_2 = S_2(x) - S_1(x)$. Esto nos da por inducción un procedimiento para extraer la base y los funcionales biortogonales: para cada número natural n , cogemos $0 \neq e_n \in S_n(X) \cap S_{n-1}^{-1}(0)$ y definimos $e_n^* : X \rightarrow \mathbb{R}$ por $e_n^*(x)e_n = S_n(x) - S_{n-1}(x)$. Entonces

$$|e_n^*(x)| = \|S_n(x) - S_{n-1}(x)\| \|e_n\|^{-1} \leq 2 \sup_n \|S_n\| \|e_n\|^{-1} \|x\|;$$

por lo tanto $e_n^* \in X^*$. Es inmediato comprobar que $e_k^*(e_j) = \delta_{kj}$ para cualesquiera números naturales k, j .

Por otro lado si $S_0(x) = 0$ para todo x , podemos escribir

$$S_n(x) = \sum_{k=1}^n (S_k(x) - S_{k-1}(x)) = \sum_{k=1}^n e_k^*(x)e_k,$$

que por la hipótesis (III) converge a x para todo $x \in X$. Entonces, la sucesión $(e_n)_{n=1}^\infty$ es una base y $(S_n)_{n=1}^\infty$ sus proyecciones naturales. \square

En la siguiente definición relajamos la suposición de que una base debe expandir el espacio entero. Vamos a introducir el concepto de *sucesión básica* el cual es de fundamental importancia en la teoría de espacios de Banach.

2.1.8 Definición. *Una sucesión $(e_k)_{k=1}^\infty$ en un espacio de Banach X es una sucesión básica si es una base para $[e_k]_{k \in \mathbb{N}}$, el cierre de la expansión lineal de $(e_k)_{k=1}^\infty$.*

Para reconocer las sucesiones de elementos en un espacio de Banach como una sucesión básica tenemos la siguiente proposición también conocida como el *Criterio de Grunblum*.

2.1.9 Proposición. Una sucesión $(e_k)_{k=1}^{\infty}$ de elementos distintos de cero de un espacio de Banach X es una sucesión básica si y solo si existe una constante positiva K tal que

$$\left\| \sum_{k=1}^m a_k e_k \right\| \leq K \left\| \sum_{k=1}^n a_k e_k \right\| \quad (2.1)$$

para cada sucesión de escalares $(a_k)_{k=1}^{\infty}$ y cualesquiera enteros m, n tales que $m \leq n$.

Demostración. Asumimos que $(e_k)_{k=1}^{\infty}$ es una sucesión básica y sean $S_m : [e_k] \rightarrow [e_k]$, $m = 1, 2, \dots$, las proyecciones de las sumas parciales. Si $m \leq n$ tenemos

$$\left\| \sum_{k=1}^m a_k e_k \right\| = \left\| S_m \left(\sum_{k=1}^n a_k e_k \right) \right\| \leq \sup_m \|S_m\| \left\| \sum_{k=1}^n a_k e_k \right\|,$$

por lo que 2.1 se da con $K = \sup_m \|S_m\|$.

Para la otra implicación, sea E la expansión lineal de $(e_k)_{k=1}^{\infty}$. La condición 2.1 implica que los vectores $(e_k)_{k=1}^{\infty}$ son linealmente independientes. Esto nos permite definir para cada m un operador de rango finito $s_m : E \rightarrow [e_k]_{k=1}^m$ por

$$s_m \left(\sum_{k=1}^n a_k e_k \right) = \sum_{k=1}^{\min(m,n)} a_k e_k, \quad m, n \in \mathbb{N}.$$

Por densidad cada s_m se extiende a $S_m : [e_k] \rightarrow [e_k]_{k=1}^m$ con $\|S_m\| = \|s_m\| \leq K$.

Nos damos cuenta de que para cada $x \in E$ tenemos

$$S_n S_m(x) = S_m S_n(x) = S_{\min(m,n)}(x), \quad m, n \in \mathbb{N}, \quad (2.2)$$

y, por tanto, 2.2 se cumple para todo $x \in [e_k]$.

Para cada $x \in [e_k]$ la sucesión $(S_n(x))_{k=1}^{\infty}$ converge a x , ya que el conjunto $\{x \in [e_k] : S_m(x) \rightarrow x\}$ es cerrado y contiene a E , el cual es denso en $[e_k]$. Ahora por la Proposición 2.1.7 $(e_k)_{k=1}^{\infty}$ es una base para $[e_k]$ con proyecciones asociadas $(S_m)_{m=1}^{\infty}$. \square

2.1.2. Ejemplos: Series de Fourier

Algunos de los espacios de Banach clásicos vienen con una base natural dada. Por ejemplo en los espacios ℓ_p para $1 \leq p < \infty$ y c_0 existe una base canónica dada por la sucesión $e_n = (0, \dots, 0, 1, 0, \dots)$ donde la única entrada distinta de 0 es la n -ésima. Vamos a mostrar un ejemplo que proviene del análisis de Fourier y también la construcción original de Schauder de una base en $C[0, 1]$.

Sea $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$. Denotamos un elemento de \mathbb{T} por $e^{i\theta}$ y podemos identificar el espacio $C_{\mathbb{C}}(\mathbb{T})$ de las funciones continuas de variable compleja en \mathbb{T} con el espacio de

las funciones 2π -periódicas continuas en \mathbb{R} . En el contexto de las series de Fourier es más natural considerar espacios de funciones complejas que de funciones reales.

Para cada $n \in \mathbb{N}$ sea $e_n \in \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ la función tal que $e_n(\theta) = e^{in\theta}$. La pregunta que queremos abordar es si la sucesión $(e_0, e_1, e_{-1}, e_2, e_{-2}, \dots)$ en este orden particular es una base de $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$. De hecho veremos que no lo es. Este es un resultado clásico en análisis de Fourier que es equivalente a decir que hay una función continua f cuya serie de Fourier no converge uniformemente.

Que $[e_n]_{n \in \mathbb{N}} = \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ se obtiene del teorema de Stone-Weierstrass aunque lo vamos a probar directamente.

Los *coeficientes de Fourier* de $f \in \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ están definidos por la fórmula

$$\hat{f}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} dt, \quad n \in \mathbb{Z}.$$

Se comprueba de manera rutinaria que los funcionales lineales

$$e_n^* : \mathcal{C}_{\mathbb{C}}(\mathbb{T}) \rightarrow \mathbb{C}, \quad f \mapsto e_n^*(f) = \hat{f}(n)$$

son biortogonales a la sucesión $(e_n)_{n \in \mathbb{Z}}$.

La *serie de Fourier* de f es la serie formal

$$\sum_{n=-\infty}^{n=\infty} \hat{f}(n) e^{in\theta}.$$

Vamos a enunciar dos lemas que nos serán muy útiles más adelante.

2.1.10 Lema. *Para todo $t \in \mathbb{R}$, se tiene*

$$(I) \quad \sin\left(\frac{t}{2}\right) \sum_{k=0}^n \cos(kt) = \cos\left(\frac{nt}{2}\right) \sin\left(\frac{(n+1)t}{2}\right),$$

$$(II) \quad \sin\left(\frac{t}{2}\right) \sum_{k=1}^n \sin(kt) = \sin\left(\frac{nt}{2}\right) \sin\left(\frac{(n+1)t}{2}\right).$$

Demostración. Vamos a probar ambas igualdades simultáneamente. Comencemos:

$$\sin\left(\frac{t}{2}\right) \sum_{k=0}^n \cos(kt) + i \sin\left(\frac{t}{2}\right) \sum_{k=0}^n \sin(kt) = \sin\left(\frac{t}{2}\right) \sum_{k=0}^n (\cos(kt) + i \sin(kt)).$$

Usamos la fórmula de Moivre y obtenemos

$$\sin\left(\frac{t}{2}\right) \sum_{k=0}^n (\cos(kt) + i \sin(kt)) = \sin\left(\frac{t}{2}\right) \sum_{k=0}^n (\cos(t) + i \sin(t))^k.$$

Sea $z = \cos(t) + i \sin(t)$. Sabemos que

$$1 + z + \cdots + z^n = \frac{1 - z^{n+1}}{1 - z}$$

con la salvedad de que $z = 1$. En ese caso $t = 2k\pi$, $k \in \mathbb{Z}$ y las igualdades son ciertas trivialmente. Usando $z^{n+1} = \cos((n+1)t) + i \sin((n+1)t)$ junto con las igualdades trigonométricas $2 \sin^2(\alpha/2) = 1 - \cos(\alpha)$ y $\sin(\alpha) = 2 \sin(\alpha/2) \cos(\alpha/2)$ llegamos a

$$\begin{aligned} & \operatorname{sen}\left(\frac{t}{2}\right) \frac{1 - \cos((n+1)t) - i \sin((n+1)t)}{1 - \cos(t) - i \sin(t)} = \\ &= \frac{2 \sin^2\left(\frac{n+1}{2}t\right) - 2i \sin\left(\frac{n+1}{2}t\right) \cos\left(\frac{n+1}{2}t\right)}{2 \sin\left(\frac{t}{2}\right) - 2i \cos\left(\frac{t}{2}\right)}. \end{aligned}$$

Simplificamos,

$$\operatorname{sen}\left(\frac{n+1}{2}t\right) \frac{\sin\left(\frac{n+1}{2}t\right) - i \cos\left(\frac{n+1}{2}t\right)}{\sin\left(\frac{t}{2}\right) - i \cos\left(\frac{t}{2}\right)}.$$

Multiplicamos numerador y denominador por el conjugado del denominador

$$\operatorname{sen}\left(\frac{n+1}{2}t\right) \left(\operatorname{sen}\left(\frac{n+1}{2}t\right) - i \cos\left(\frac{n+1}{2}t\right) \right) \left(\operatorname{sen}\left(\frac{t}{2}\right) + i \cos\left(\frac{t}{2}\right) \right).$$

Desarrollando la parte real y usando el coseno de la diferencia de dos ángulos

$$\begin{aligned} & \operatorname{sen}\left(\frac{n+1}{2}t\right) \left(\cos\left(\frac{n+1}{2}t\right) \cos\left(\frac{t}{2}\right) + \operatorname{sen}\left(\frac{n+1}{2}t\right) \operatorname{sen}\left(\frac{t}{2}\right) \right) = \\ &= \operatorname{sen}\left(\frac{n+1}{2}t\right) \cos\left(\frac{n+1}{2}t - \frac{t}{2}\right) = \operatorname{sen}\left(\frac{n+1}{2}t\right) \cos\left(\frac{nt}{2}\right). \end{aligned}$$

A continuación desarrollamos la parte imaginaria y usamos el seno de la diferencia

$$\begin{aligned} & \operatorname{sen}\left(\frac{n+1}{2}t\right) \left(\left(\operatorname{sen}\left(\frac{n+1}{2}t\right) \cos\left(\frac{t}{2}\right) - \cos\left(\frac{n+1}{2}t\right) \operatorname{sen}\left(\frac{t}{2}\right) \right) \right) = \\ &= \operatorname{sen}\left(\frac{n+1}{2}t\right) \operatorname{sen}\left(\frac{n+1}{2}t - \frac{t}{2}\right) = \operatorname{sen}\left(\frac{n+1}{2}t\right) \operatorname{sen}\left(\frac{nt}{2}\right). \end{aligned}$$

Finalizamos la demostración juntando ambas partes,

$$\operatorname{sen}\left(\frac{n+1}{2}t\right) \left(\cos\left(\frac{nt}{2}\right) + i \operatorname{sen}\left(\frac{nt}{2}\right) \right).$$

□

2.1.11 Lema. Sean $n \in \mathbb{N}$ y $t \in \mathbb{R}$. La siguiente igualdad es cierta:

$$\operatorname{sen}\left(\frac{t}{2}\right) \sum_{k=-n}^n e^{ikt} = \operatorname{sen}\left(\left(n + \frac{1}{2}\right)t\right).$$

Demostración. Vamos a desarrollar el sumatorio y seguidamente podremos utilizar la paridad del coseno y la imparidad del seno

$$\begin{aligned} \sum_{k=-n}^n e^{ikt} &= \sum_{k=-n}^n (\cos(kt) + i \sin(kt)) = \sum_{k=-n}^n \cos(kt) + i \sum_{k=-n}^n \sin(kt) \\ &= \sum_{k=-n}^n \cos(kt) = 2 \sum_{k=1}^n \cos(kt) + 1 = 2 \sum_{k=0}^n \cos(kt) - 1. \end{aligned}$$

Usando lo anterior y aplicando el apartado (I) del Lema 2.1.10 obtenemos

$$\begin{aligned} \operatorname{sen}\left(\frac{t}{2}\right) \sum_{k=-n}^n e^{ikt} &= 2 \operatorname{sen}\left(\frac{t}{2}\right) \sum_{k=0}^n \cos(kt) - \operatorname{sen}\left(\frac{t}{2}\right) \\ &= 2 \cos\left(\frac{nt}{2}\right) \operatorname{sen}\left(\frac{(n+1)t}{2}\right) - \operatorname{sen}\left(\frac{t}{2}\right). \end{aligned}$$

Desarrollando y usando la fórmula para el seno de la suma de dos ángulos:

$$\begin{aligned} 2 \cos\left(\frac{nt}{2}\right) \operatorname{sen}\left(\frac{nt}{2} + \frac{t}{2}\right) - \operatorname{sen}\left(\frac{t}{2}\right) &= \\ = 2 \cos\left(\frac{nt}{2}\right) \left[\operatorname{sen}\left(\frac{nt}{2}\right) \cos\left(\frac{t}{2}\right) + \cos\left(\frac{nt}{2}\right) \operatorname{sen}\left(\frac{t}{2}\right) \right] - \operatorname{sen}\left(\frac{t}{2}\right) &= \\ = \operatorname{sen}(nt) \cos\left(\frac{t}{2}\right) + 2 \cos\left(\frac{nt}{2}\right)^2 \operatorname{sen}\left(\frac{t}{2}\right) - \operatorname{sen}\left(\frac{t}{2}\right). & \end{aligned}$$

Recordamos que $2 \cos^2(\alpha/2) = \cos(\alpha) + 1$ para cualquier $\alpha \in \mathbb{R}$ y posteriormente identificamos el seno de la suma en la expresión finalizando la demostración.

$$\begin{aligned} \operatorname{sen}(nt) \cos\left(\frac{t}{2}\right) + (\cos(nt) + 1) \operatorname{sen}\left(\frac{t}{2}\right) - \operatorname{sen}\left(\frac{t}{2}\right) &= \\ = \operatorname{sen}(nt) \cos\left(\frac{t}{2}\right) + \cos(nt) \operatorname{sen}\left(\frac{t}{2}\right) &= \\ = \operatorname{sen}\left(nt + \frac{t}{2}\right) &= \operatorname{sen}\left(\left(n + \frac{1}{2}\right)t\right). \end{aligned}$$
□

Para cada entero n sea $T_n : \mathcal{C}_{\mathbb{C}}(\mathbb{T}) \rightarrow \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ el operador

$$T_n(f) = \sum_{k=-n}^n \hat{f}(k) e_k,$$

nos da la n -ésima suma parcial de la serie de Fourier de f . Entonces

$$\begin{aligned} T_n(f)(\theta) &= \sum_{k=-n}^n \frac{1}{2\pi} \int_{-\pi}^{\theta+\pi} f(t) e^{ik(\theta-t)} dt = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta-t) \sum_{k=-n}^n e^{ikt} dt \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta-t) \frac{\operatorname{sen}(n + \frac{1}{2})t}{\operatorname{sen}\frac{t}{2}} dt. \end{aligned}$$

En la última igualdad hemos usado el Lema 2.1.11. La función

$$D_n(t) = \frac{\sin(n + \frac{1}{2})t}{\sin \frac{t}{2}}$$

es conocida como el *núcleo de Dirichlet*.

Consideremos también los operadores

$$A_n = \frac{1}{n}(T_0 + \cdots + T_{n-1}), \quad n = 2, 3, \dots \quad (2.3)$$

Entonces

$$A_n f(\theta) = \frac{1}{2\pi n} \int_{-\pi}^{\pi} f(\theta - t) \sum_{k=0}^{n-1} \frac{\sin(n + \frac{1}{2})t}{\sin \frac{t}{2}} dt = \frac{1}{2\pi n} \int_{-\pi}^{\pi} f(\theta - t) \left(\frac{\sin \frac{nt}{2}}{\sin \frac{t}{2}} \right)^2 dt.$$

La función

$$F_n(t) = \frac{1}{n} \left(\frac{\sin \frac{nt}{2}}{\sin \frac{t}{2}} \right)^2$$

es el *núcleo de Fejér*. En el siguiente lema mostramos una propiedad que verifican ambos núcleos.

2.1.12 Lema. Sea $n \in \mathbb{N}$, entonces

$$(I) \quad \frac{1}{2\pi} \int_{-\pi}^{\pi} D_n(t) dt = 1,$$

$$(II) \quad \frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(t) dt = 1.$$

Demostración. (I) De la demostración del Lema 2.1.11 es claro ver la caracterización $D_n(t) = 1 + 2 \sum_{k=1}^n \cos(kt)$. Integraremos usándola:

$$\begin{aligned} \frac{1}{2\pi} \int_{-\pi}^{\pi} D_n(t) dt &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(1 + \sum_{k=1}^n \cos(kt) \right) dt = 1 + \frac{1}{2\pi} \sum_{k=1}^n \int_{-\pi}^{\pi} \cos(kt) dt = \\ &= 1 + \frac{1}{2\pi} \sum_{k=1}^n \frac{2 \sin(k\pi)}{k} = 1. \end{aligned}$$

(II) Sabemos que

$$A_n f(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta - t) F_n(t) dt,$$

y tomando f como la función identidad nos queda

$$A_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(t) dt \quad \text{pero} \quad A_n = \frac{T_0 + \cdots + T_{n-1}}{n}$$

por la ecuación 2.3. Calculamos los coeficientes de Fourier de f que son $a_0 = 2$ y $a_n = b_n = 0$ para $n \geq 1$ de manera trivial. Ya hemos finalizado la demostración pues

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(t) dt = \frac{T_0 + \dots + T_{n-1}}{n} = \frac{n \frac{a_0}{2}}{n} = 1. \quad \square$$

Sin embargo, una diferencia crucial es que F_n es una función positiva mientras que D_n no lo es.

Mostremos ahora que si $f \in \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ entonces $\|A_n f - f\| \rightarrow 0$. Como f es uniformemente continua, dado $\varepsilon > 0$ podemos encontrar un $0 < \delta < \pi$ de manera que $|\theta - \theta'| < \delta$ implica que $|f(\theta) - f(\theta')| \leq \varepsilon$. Entonces para cada θ tenemos

$$A_n f(\theta) - f(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_n(t)(f(\theta - t) - f(\theta))dt.$$

Por tanto

$$\|A_n f - f\| \leq \frac{1}{2\pi} \|f\| \int_{\delta < |t| \leq \pi} F_n(t) dt + \frac{1}{2\pi} \varepsilon \int_{-\delta}^{\delta} F_n(t) dt.$$

Ahora

$$\frac{1}{2\pi} \|f\| \int_{\delta < |t| \leq \pi} F_n(t) dt \leq \frac{1}{n} \operatorname{sen}^{-2} \frac{\delta}{2}$$

y por tanto

$$\limsup \|A_n f - f\| \leq \varepsilon.$$

Esto muestra que $[e_n]_{n \in \mathbb{Z}} = \mathcal{C}_{\mathbb{C}}(\mathbb{T})$.

Como los funcionales biortogonales están dados por los coeficientes de Fourier, se sigue que si $(e_0, e_1, e_{-1}, \dots)$ es una base entonces los operadores de las sumas parciales $(S_n)_{n=1}^{\infty}$ satisfacen $S_{2n+1} = T_n$ para todo \mathbb{N} . Para mostrar que no es una base es suficiente ver que la sucesión de operadores $(T_n)_{n=1}^{\infty}$ no está uniformemente acotada.

Sea $\varphi \in \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ dado por

$$\varphi(f) = f(0).$$

Entonces

$$\varphi(T_n f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} D_n(t) f(-t) dt;$$

por tanto

$$\|T_n^* \varphi\| = \frac{1}{2\pi} \int_{-\pi}^{\pi} |D_n(t)| dt.$$

De esta forma, como $|\operatorname{sen} x| \leq |x|$ para todo x real,

$$\begin{aligned}\|\mathbf{T}_n\| &\geq \frac{1}{2\pi} \int_{-\pi}^{\pi} |\mathbf{D}_n(t)| dt = \frac{1}{\pi} \int_0^\pi \left| \frac{\operatorname{sen}(n + \frac{1}{2})t}{\operatorname{sen} \frac{t}{2}} \right| dt \\ &\geq \frac{2}{\pi} \int_0^{(n+1/2)\pi} \left| \frac{\operatorname{sen} t}{\operatorname{sen} \frac{t}{2n+1}} \right| \frac{dt}{2n+1} \geq \frac{2}{\pi} \int_0^{(n+1/2)\pi} \left| \frac{\operatorname{sen} t}{t} \right| dt.\end{aligned}$$

Por lo tanto,

$$\liminf_{n \rightarrow \infty} \|\mathbf{T}_n\| \geq \frac{2}{\pi} \int_0^\infty \frac{|\operatorname{sen} x|}{x} dx = \infty.$$

Ya hemos probado que $\sup_n \|\mathbf{T}_n^* \varphi\| = \infty$; entonces por el principio de acotación de uniforme existe $f \in \mathcal{C}_{\mathbb{C}}(\mathbb{T})$ tal que $(\mathbf{T}_n f(0))_{n=1}^\infty$ es no acotado.

Si preferimos tratar con el espacio de las funciones continuas de variable real $\mathcal{C}(\mathbb{T})$, exactamente los mismos cálculos muestran que el sistema trigonométrico (real)

$$\{1, \cos \theta, \operatorname{sen} \theta, \cos 2\theta, \operatorname{sen} 2\theta, \dots\}$$

no es una base. En efecto, los operadores $(\mathbf{T}_n)_{n=1}^\infty$ están no acotados en el espacio $\mathcal{C}(\mathbb{T})$ y corresponden a los operadores de las sumas parciales $(S_{2n+1})_{n=1}^\infty$ como antes.

Sin embargo, $\mathcal{C}(\mathbb{T})$ y $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$ tienen una base. Podemos mostrar esto de manera similar a la construcción original de la base de Shauder en $\mathcal{C}[0, 1]$ que ahora describiremos. Sea $(q_n)_{n=1}^\infty$ una sucesión que es densa en $[0, 1]$ y tal que $q_1 = 0$ y $q_2 = 1$. Construimos inductivamente una sucesión de operadores $(S_n)_{n=1}^\infty$, definidos en $\mathcal{C}[0, 1]$, por $S_1 f(t) = f(q_1)$ para $0 \leq t \leq 1$, y posteriormente $S_n f$ es la función lineal a trozos definida por $S_n f(q_k) = f(q_k)$ para $1 \leq k \leq n$ y lineal en todos los intervalos de $[0, 1] \setminus \{q_1, q_2, \dots, q_n\}$. Es fácil ver que $\|S_n\| = 1$ para todo n y que se cumplen todas las hipótesis de la Proposición 2.1.7. De esta manera obtenemos una base monótona para $\mathcal{C}[0, 1]$. Los elementos de la base están dados por $e_1(t) = 1$ para todo t , y entonces e_n se define recursivamente por $e_n(q_n) = 1$, $e_n(q_k) = 0$ para $1 \leq k \leq n-1$, y e_n es lineal en cada intervalo en $[0, 1] \setminus \{q_1, q_2, \dots, q_n\}$.

Para modificar esto para el caso del caso del círculo identificamos $\mathcal{C}(\mathbb{T})$ (respectivamente $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$) con las funciones en $\mathcal{C}[0, 2\pi]$ (respectivamente $\mathcal{C}_{\mathbb{C}}[0, 2\pi]$) tal que $f(0) = f(2\pi)$. Sea $q_1 = 0$ y supongamos $(q_n)_{n=1}^\infty$ es denso en $[0, 2\pi]$. Entonces definimos $S_n f$ para $n > 1$ por $S_n f(q_k) = f(q_k)$ para $1 \leq k \leq n$ y $S_n f(2\pi) = f(q_1)$ y para ser afín en cada intervalo en $[0, 2\pi] \setminus \{q_1, \dots, q_n\}$.

En ambos casos este procedimiento construye una base monótona. Resumimos esto en el siguiente teorema.

2.1.13 Teorema. *Los espacios $\mathcal{C}[0, 1]$ y $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$ tienen una base monótona. El sistema trigonométrico complejo $(1, e^{i\theta}, e^{-i\theta}, \dots)$ no es una base de $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$.*

2.1.3. Equivalencia de bases y sucesiones básicas

Cuando seleccionamos una base de un espacio de vectores de dimensión finita estamos, en efecto, seleccionando un sistema de coordenadas. Las bases en espacios de Banach de dimensión infinita juegan el mismo papel. Así, si tenemos una base $(e_n)_{n=1}^{\infty}$ entonces podemos escribir x usando sus coordenadas $(e_n^*(x))_{n=1}^{\infty}$. Por supuesto no es cierto que toda sucesión de escalares $(a_n)_{n=1}^{\infty}$ define un elemento de X . Por tanto obtenemos un sistema de coordenadas de X a través de una cierta sucesión del espacio, es decir, un subespacio lineal del espacio de vectores de todas las sucesiones. Esto nos lleva naturalmente a la siguiente definición.

2.1.14 Definición. *Dos bases (o sucesiones básicas) $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ en sus respectivos espacios de Banach X e Y son equivalentes si siempre que tomamos una sucesión de escalares $(a_n)_{n=1}^{\infty}$, entonces $\sum_{n=1}^{\infty} a_n x_n$ converge si y solo si $\sum_{n=1}^{\infty} a_n y_n$ converge.*

Por lo tanto si las bases $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ son equivalentes entonces los espacios de sucesiones asociados a X por $(x_n)_{n=1}^{\infty}$ y a Y por $(y_n)_{n=1}^{\infty}$ coinciden. Es una consecuencia fácil del teorema de la gráfica cerrada que si $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ son equivalentes, entonces los espacios X e Y deben ser isomorfos. De una manera más precisa tenemos el siguiente teorema.

2.1.15 Teorema. *Dos bases (o sucesiones básicas) son equivalentes si y solo si existe un isomorfismo $T : [x_n] \rightarrow [y_n]$ de manera que $Tx_n = y_n$ para cada $n \in \mathbb{N}$.*

Demostración. Sea $X = [x_n]$ e $Y = [y_n]$. Es obvio que $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ son equivalentes si existe un isomorfismo T de X a Y tal que $Tx_n = y_n$ para cada $n \in \mathbb{N}$.

Supongamos de manera inversa que $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ son equivalentes. Definamos $T : X \rightarrow Y$ por $T(\sum_{n=1}^{\infty} a_n x_n) = \sum_{n=1}^{\infty} a_n y_n$. Entonces T es inyectiva y sobreyectiva. Para probar que es continua usaremos el teorema de la gráfica cerrada. Supongamos $(u_j)_{j=1}^{\infty}$ es una sucesión tal que $u_j \rightarrow u$ en X y $Tu_j \rightarrow v$ en Y . $u_j = \sum_{n=1}^{\infty} x_n^*(u_j)x_n$ y $u = \sum_{n=1}^{\infty} x_n^*(u)x_n$. Se sigue de la continuidad de los funcionales biortogonales asociados respectivamente a $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ que $x_n^*(u_j) \rightarrow x_n^*(u)$ e $y_n^*(Tu_j) = x_n^*(u_j) \rightarrow y_n^*(v)$ para todo n . Por la unicidad del límite, $x_n^*(u) = y_n^*(v)$ para todo n . Por tanto $Tu = v$ por lo que T es continuo. \square

2.1.16 Corolario. *Sean $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ dos bases para los espacios de Banach X e Y respectivamente. Entonces $(x_n)_{n=1}^{\infty}$ es equivalente a $(y_n)_{n=1}^{\infty}$ si y solo si existe una constante $C > 0$ tal que para toda sucesión de escalares distintos de cero $(a_i)_{i=1}^{\infty}$ tenemos*

$$C^{-1} \left\| \sum_{i=1}^{\infty} a_i y_i \right\| \leq \left\| \sum_{i=1}^{\infty} a_i x_i \right\| \leq C \left\| \sum_{i=1}^{\infty} a_i y_i \right\|. \quad (2.4)$$

Si $C = 1$ en 2.4 entonces las sucesiones básicas $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ se dicen *isométricamente equivalentes*.

La equivalencia de sucesión básicas (y en particular de bases) es una técnica poderosa para estudiar la estructura isomórfica de los espacios de Banach. Introducimos ahora un tipo especial de sucesión básica.

2.1.17 Definición. Sea $(e_n)_{n=1}^{\infty}$ una base para un espacio de Banach X . Supongamos que $(p_n)_{n=1}^{\infty}$ es una sucesión estrictamente creciente de enteros con $p_0 = 0$ y que $(a_n)_{n=1}^{\infty}$ son escalares. Entonces a una sucesión de vectores distintos de cero $(u_n)_{n=1}^{\infty}$ en X de la forma

$$u_n = \sum_{j=p_{n-1}+1}^{p_n} a_j e_j.$$

la llamaremos *sucesión básica por bloques de $(e_n)_{n=1}^{\infty}$* .

2.1.18 Lema. Supongamos $(e_n)_{n=1}^{\infty}$ es una base para el espacio de Banach X con constante básica K_b . Sea $(u_k)_{k=1}^{\infty}$ una sucesión básica por bloques de $(e_n)_{n=1}^{\infty}$. Entonces $(u_k)_{k=1}^{\infty}$ es una sucesión básica con constante básica menor o igual a K_b .

Demostración. Supongamos que $u_k = \sum_{j=p_{k-1}+1}^{p_k} a_j e_j$, $k \in \mathbb{N}$, es una sucesión básica por bloques de $(e_n)_{n=1}^{\infty}$. Entonces para cualesquiera escalares (b_k) y enteros m, n con $m \leq n$ tenemos

$$\begin{aligned} \left\| \sum_{k=1}^m b_k u_k \right\| &= \left\| \sum_{k=1}^m b_k \sum_{j=p_{k-1}+1}^{p_k} a_j e_j \right\| = \left\| \sum_{k=1}^m \sum_{j=p_{k-1}+1}^{p_k} b_k a_j e_j \right\| \\ &= \left\| \sum_{j=1}^{p_m} c_j e_j \right\|, \text{ donde } c_j = a_j b_k \text{ si } p_{k-1} + 1 \leq j \leq p_k \\ &\leq K_b \left\| \sum_{j=1}^{p_n} c_j e_j \right\| = K_b \left\| \sum_{k=1}^n b_k u_k \right\|. \end{aligned}$$

Esto es, $(u_k)_{k=1}^{\infty}$ satisface la condición de Grunblum (Proposición 2.1.9) por lo que $(u_k)_{k=1}^{\infty}$ es una sucesión básica con constante básica como mucho K_b . \square

2.1.19 Definición. Una sucesión básica $(x_n)_{n=1}^{\infty}$ en X está complementada si $[x_n]$ es un subespacio complementado de X .

2.1.20 Nota. Supongamos que $(x_n)_{n=1}^{\infty}$ es una sucesión básica complementada en un espacio de Banach X . Sea $Y = [x_n]$ y $P : X \rightarrow Y$ una proyección. Si $(x_n^*)_{n=1}^{\infty} \subset Y^*$ son los funcionales biortogonales asociados a $(x_n)_{n=1}^{\infty}$, entonces usando el teorema de Hahn-Banach podemos obtener una sucesión biortogonal $(\hat{x}_n^*)_{n=1}^{\infty} \subset X^*$ tal que cada \hat{x}_n^* es una

extensión x_n^* que preserva la norma. Pero ya que tenemos una proyección, P , podemos también extender cada x_n^* a todo el espacio X poniendo $u_n^* = x_n^* \circ P$. Entonces para cada $x \in X$ tendremos

$$\sum_{n=1}^{\infty} u_n^*(x) x_n = P(x)$$

De manera la inversa, si podemos construir una sucesión $(u_n^*)_{n=1}^{\infty} \subset X^*$ tal que $u_n^*(x_m) = \delta_{nm}$ y la serie $\sum_{n=1}^{\infty} u_n^*(x) x_n$ converge $\forall x \in X$, entonces el subespacio $[x_n]$ está complementado por la proyección $X \rightarrow [x_n]$, $x \mapsto \sum_{n=1}^{\infty} u_n^*(x) x_n$.

2.1.21 Definición. Sean X e Y dos espacios de Banach. Decimos que dos sucesiones $(x_n)_{n=1}^{\infty} \subset X$ e $(y_n)_{n=1}^{\infty} \subset Y$ son congruentes con respecto a (X, Y) si existe un operador $T : X \rightarrow Y$ tal que $T(x_n) = y_n$ para todo $n \in \mathbb{N}$. Cuando (x_n) e (y_n) satisfacen esta condición en el caso particular de $X = Y$, simplemente diremos que son congruentes.

Supongamos que las sucesiones $(x_n)_{n=1}^{\infty} \subset X$ e $(y_n)_{n=1}^{\infty} \subset Y$ son congruentes con respecto a (X, Y) . El operador T de X a Y que existe por la definición previa preserva todas las propiedades isomórficas de $(x_n)_{n=1}^{\infty}$. Por ejemplo, si $(x_n)_{n=1}^{\infty}$ es una base de X , entonces $(y_n)_{n=1}^{\infty}$ es una base de Y ; si K_b es la constante básica de $(x_n)_{n=1}^{\infty}$, entonces la constante básica de $(y_n)_{n=1}^{\infty}$ es $K_b \|T\| \|T^{-1}\|$.

El siguiente resultado de estabilidad data de 1940. Más o menos, dice que si $(x_n)_{n=1}^{\infty}$ es una sucesión básica en un espacio de Banach X e $(y_n)_{n=1}^{\infty}$ es otra sucesión en X tal que $(\|x_n - y_n\|)_{n=1}^{\infty}$ converge suficientemente rápido a 0, entonces $(y_n)_{n=1}^{\infty}$ y $(x_n)_{n=1}^{\infty}$ son congruentes.

2.1.22 Teorema. (*Principio de las pequeñas perturbaciones*). Sea $(x_n)_{n=1}^{\infty}$ una base en un espacio de Banach X con constante básica K_b . Si $(y_n)_{n=1}^{\infty}$ es una sucesión en X tal que

$$2K_b \sum_{n=1}^{\infty} \frac{\|x_n - y_n\|}{\|x_n\|} = \theta < 1,$$

entonces $(x_n)_{n=1}^{\infty}$ e $(y_n)_{n=1}^{\infty}$ son congruentes. En particular:

- (I) $(y_n)_{n=1}^{\infty}$ es una base con constante básica como mucho $K_b(1 + \theta)(1 - \theta)^{-1}$.
- (II) Si $(x_n)_{n=1}^{\infty}$ es una base, también lo es $(y_n)_{n=1}^{\infty}$.
- (III) Si $[x_n]$ está complementado en X , también lo está $[y_n]$.

Demostración. Para $n \geq 2$ y $x \in [x_n]$ tenemos

$$x_n^*(x) x_n = \sum_{k=1}^n x_k^*(x) x_k - \sum_{k=1}^{n-1} x_k^*(x) x_k,$$

donde $(x_n^*)_{n=1}^\infty \subset [x_n]^*$ son los funcionales biortogonales de $(x_n)_{n=1}^\infty$. Entonces

$$\|x_n^*(x)x_n\| \leq 2K_b \|x\|,$$

y entonces

$$\|x_n^*\| \|x_n\| \leq 2K_b.$$

Para $n = 1$ es claro que $\|x_1^*\| \|x_1\| \leq K_b$. Estas desigualdades se mantienen si reemplazamos x_n^* por su extensión Hahn-Banach \hat{x}_n^* de X .

Para cada $x \in X$ ponemos

$$T(x) = x + \sum_{n=1}^{\infty} \hat{x}_n^*(x)(y_n - x_n).$$

Entonces T es un operador acotado de X en X con $T(x_n) = y_n$ y con norma

$$\|T\| \leq 1 + \sum_{n=1}^{\infty} \|\hat{x}_n^*\| \|y_n - x_n\| \leq 1 + 2K_b \sum_{n=1}^{\infty} \frac{\|y_n - x_n\|}{\|x_n\|} = 1 + \theta.$$

Más aún,

$$\|T - I_X\| \leq \sum_{n=1}^{\infty} \|\hat{x}_n^*\| \|y_n - x_n\| = \theta < 1,$$

lo cual implica que T es invertible y $\|T^{-1}\| \leq (1 - \theta)^{-1}$. \square

2.2. Tipos especiales de bases

Saber si un espacio de Banach separable tiene una base e identificar una que permita calcular fácilmente la norma de sus elementos es importante. Sin embargo, este conocimiento pasa a ser muy limitado si estamos interesados en usar las bases como una herramienta para profundizar en la geometría del espacio. En este apartado miraremos de una forma más cuidadosa los tipos espaciales de bases. En particular, consideraremos las nociones de bases shrinking, acotadamente completas, incondicionales y simétricas.

2.2.1. Bases incondicionales

Las bases incondicionales son muy útiles y son las más extensamente estudiadas debido a las buenas propiedades estructurales de los espacios que abarcan. Las bases incondicionales aparecieron por primera vez en 1948 en el trabajo de Karlin [4], quien demostró que $C[0, 1]$ no puede tener una base incondicional. Probaremos a lo largo del subapartado este hecho.

2.2.1 Definición. Una base $(u_n)_{n=1}^{\infty}$ de un espacio de Banach X es *incondicional* si para cada $x \in X$ la serie $\sum_{n=1}^{\infty} u_n^*(x)u_n$ converge incondicionalmente.

Obviamente, $(u_n)_{n=1}^{\infty}$ es una base incondicional de X si y solo si $(u_{\pi(n)})_{n=1}^{\infty}$ es una base de X para toda permutación $\pi : \mathbb{N} \rightarrow \mathbb{N}$.

2.2.2 Ejemplo. La base de vectores unitarios estándar $(e_n)_{n=1}^{\infty}$ es una base incondicional de c_0 y ℓ_p para $1 \leq p < \infty$. Un ejemplo de una base que es condicional (i.e., no es incondicional) es la *base sumante* de c_0 , definida como

$$f_n = e_1 + \cdots + e_n, \quad n \in \mathbb{N}.$$

Para ver que $(f_n)_{n=1}^{\infty}$ es una base de c_0 probamos que para cada $\xi = (\xi(n))_{n=1}^{\infty} \in c_0$ tenemos $\xi = \sum_{n=1}^{\infty} f_n^*(\xi)f_n$, donde $f_n^* = e_n^* - e_{n+1}^*$ son los funcionales biortogonales de $(f_n)_{n=1}^{\infty}$. Dado $N \in \mathbb{N}$,

$$\begin{aligned} \sum_{n=1}^N f_n^*(\xi)f_n &= \sum_{n=1}^N (e_n^*(\xi) - e_{n+1}^*(\xi))f_n = \sum_{n=1}^N (\xi(n) - \xi(n+1))f_n \\ &= \sum_{n=1}^N \xi(n)f_n - \sum_{n=2}^{N+1} \xi(n)f_{n-1} = \sum_{n=1}^N \xi(n)(f_n - f_{n-1}) - \xi(N+1)f_N \\ &= \sum_{n=1}^N \xi(n)e_n - \xi(N+1)f_N, \end{aligned}$$

donde hemos asumido que $f_0 = 0$. Por tanto,

$$\begin{aligned} \left\| \xi - \sum_{n=1}^N f_n^*(\xi)f_n \right\|_{\infty} &= \left\| \sum_{N+1}^{\infty} \xi(n)e_n + \xi(N+1)f_N \right\|_{\infty} \\ &\leq \left\| \sum_{N+1}^{\infty} \xi(n)e_n \right\|_{\infty} + |\xi(N+1)| \|f_N\|_{\infty} \xrightarrow{N \rightarrow \infty} 0, \end{aligned}$$

y $(f_n)_{n=1}^{\infty}$ es una base.

Ahora identificaremos el conjunto S de coeficientes $(\alpha_n)_{n=1}^{\infty}$ tales que la serie $\sum_{n=1}^{\infty} \alpha_n f_n$ converge. De hecho, tenemos que $(\alpha_n)_{n=1}^{\infty} \in S$ si y solo si existen $\xi = (\xi(n))_{n=1}^{\infty} \in c_0$ tal que $\alpha_n = \xi(n) - \xi(n+1)$ para todo n . Entonces, claramente, a menos que las series $\sum_{n=1}^{\infty} \alpha_n$ converjan absolutamente, la convergencia de $\sum_{n=1}^{\infty} \alpha_n f_n$ en c_0 no es equivalente a la convergencia de $\sum_{n=1}^{\infty} \varepsilon_n \alpha_n f_n$ para todas las elecciones de signos $(\varepsilon_n)_{n=1}^{\infty}$. Por lo tanto $(f_n)_{n=1}^{\infty}$ no puede ser incondicional.

2.2.3 Lema. Dada una serie $\sum_{n=1}^{\infty} x_n$ en un espacio de Banach X , los siguientes enunciados son equivalentes:

- (a) $\sum_{n=1}^{\infty} x_n$ es incondicionalmente convergente (i.e., $\sum_{n=1}^{\infty} x_{\pi(n)}$ converge para toda permutación π de \mathbb{N}).
- (b) La serie $\sum_{n=1}^{\infty} x_{n_k}$ converge para toda sucesión creciente $(n_k)_{k=1}^{\infty}$.
- (c) La serie $\sum_{n=1}^{\infty} \varepsilon_n x_n$ converge para cualquier elección de signos $(\varepsilon)_{n=1}^{\infty}$.
- (d) Para cada $\varepsilon > 0$ existe n tal que si F es cualquier subconjunto finito de $\{n+1, n+2, \dots\}$, entonces

$$\left\| \sum_{j \in F} x_j \right\| < \varepsilon.$$

Demostración. (a) \Rightarrow (d) Si (d) no es cierto entonces existe $\varepsilon > 0$ tal que para cada n podemos encontrar un subconjunto finito F_n de $\{n+1, n+2, \dots\}$ con

$$\left\| \sum_{j \in F_n} x_j \right\| \geq \varepsilon.$$

Vamos a construir una permutación π de \mathbb{N} tal que $\sum_{n=1}^{\infty} x_{\pi(n)}$ diverja.

En primer lugar tomamos $n_1 = 1$ y sea $A_1 = F_{n_1}$. Después cogemos $n_2 = \max A_1$ y sea $B_1 = \{n_1 + 1, \dots, n_2\} \setminus A_1$. Repetimos el proceso y tomamos $A_2 = F_{n_2}$, $n_3 = \max A_2$ y $B_2 = \{n_2 + 1, \dots, n_3\} \setminus A_2$. Iterando, construimos una sucesión $(n_k)_{k=1}^{\infty}$ y una partición $\{n_k + 1, \dots, n_{k+1}\} = A_k \cup B_k$. Definimos π tal que π permute los elementos de $\{n_k + 1, \dots, n_{k+1}\}$ de manera que A_k preceda a B_k . Entonces la serie $\sum_{n=1}^{\infty} x_{\pi(n)}$ es divergente porque no es de Cauchy.

(b) \Rightarrow (d) Supongamos que (d) no es cierto. Entonces existen $\varepsilon > 0$ y F_n un subconjunto finito de \mathbb{N} tal que:

$$M_n = \max\{F_n : n \in \mathbb{N}\} < m_{n+1} = \min\{F_{n+1} : n \in \mathbb{N}\},$$

y $\left\| \sum_{j \in F_n} x_j \right\| \geq \varepsilon$ para todo $n \in \mathbb{N}$. Luego $F = \bigcup_{n=1}^{\infty} F_n$ es una sucesión de enteros para los cuales $\sum_{n \in F} x_n$ no converge.

(d) \Rightarrow (a) y (d) \Rightarrow (b) Por ser (d) cierto las series de (a) y (b) son de Cauchy y por ser X un espacio de Banach toda sucesión de Cauchy converge.

(a) \Leftrightarrow (c) Toda serie de números reales es incondicionalmente convergente si y solo si es absolutamente convergente. Como $\sum_{n=1}^{\infty} \|\varepsilon_n x_n\| = \sum_{n=1}^{\infty} \|x_n\|$ hemos acabado. \square

2.2.4 Definición. Una serie $\sum_{n=1}^{\infty} x_n$ en un espacio de Banach X es débilmente incondicionalmente de Cauchy (abreviadamente WUC) o débilmente incondicionalmente convergente si $\sum_{n=1}^{\infty} |x^*(x_n)| < \infty$ para todo $x^* \in X^*$.

2.2.5 Proposición. Supongamos que la serie $\sum_{n=1}^{\infty} x_n$ converge incondicionalmente a algún x en un espacio de Banach X . Entonces $\sum_{n=1}^{\infty} x_n$ es WUC.

Demostración. Dado $x^* \in X^*$, la serie de escalares $\sum_{n=1}^{\infty} x^*(x_{\pi(n)})$ converge para cada permutación π . El teorema clásico de Riemann nos dice que para sucesiones de escalares la serie $\sum_{n=1}^{\infty} a_n$ converge incondicionalmente si y solo si converge absolutamente, i.e., $\sum_{n=1}^{\infty} |a_n| < \infty$. Así tenemos $\sum_{n=1}^{\infty} |x^*(x_n)| < \infty$. \square

2.2.6 Ejemplo. La serie $\sum_{n=1}^{\infty} e_n$ en c_0 , donde $(e_n)_{n=1}^{\infty}$ es la base canónica del espacio, es WUC.

En la Proposición 2.2.8 probaremos que las series WUC están en correspondencia natural con los operadores acotados de c_0 . Veamos primero un lema.

2.2.7 Lema. Sea $\sum_{n=1}^{\infty} x_n$ una serie formal en un espacio de Banach X . La serie $\sum_{n=1}^{\infty} x_n$ es WUC si y solo si existe un $C > 0$ tal que para toda $(\xi)_{n=1}^{\infty} \in c_{00}$, tenemos

$$\left\| \sum_{n=1}^{\infty} \xi(n) x_n \right\| \leq C \max |\xi(n)|.$$

Demostración. Sea

$$S = \left\{ \sum_{n=1}^{\infty} \xi(n) x_n \in X : \xi = (\xi(n)) \in c_{00}, \|\xi\|_{\infty} \leq 1 \right\}.$$

La propiedad WUC implica que S es débilmente acotado. Entonces, está acotado para la norma por el principio de acotación uniforme.

Probamos ahora la implicación hacia la izquierda. Dado $x^* \in X^*$, sea $\varepsilon_n = \operatorname{sgn} x^*(x_n)$ que está en c_{00} . Entonces para cada N natural tenemos

$$\sum_{n=1}^N |x^*(x_n)| = \left| x^* \left(\sum_{n=1}^N \varepsilon_n x_n \right) \right| \leq C \|x^*\|,$$

y por tanto la serie $\sum_{n=1}^{\infty} |x^*(x_n)|$ converge. \square

2.2.8 Proposición. Sea $\sum_{n=1}^{\infty} x_n$ una serie en un espacio de Banach X . Entonces $\sum_{n=1}^{\infty} x_n$ es WUC si y solo si existe un operador acotado $T : c_0 \rightarrow X$ con $T e_n = x_n$.

Demostración. Si $\sum_{n=1}^{\infty} x_n$ es WUC, entonces el operador $T : c_{00} \rightarrow X$ definido por $T\xi = \sum_{n=1}^{\infty} \xi(n) x_n$ está acotado para la norma de c_0 por el Lema 2.2.7. Por densidad, T se extiende a un operador acotado $T : c_0 \rightarrow X$.

Para la otra implicación sea $T : c_0 \rightarrow X$ un operador acotado con $T e_n = x_n$ para todo n . Para cada $x^* \in X^*$ tenemos

$$\sum_{n=1}^{\infty} |x^*(x_n)| = \sum_{n=1}^{\infty} |x^*(T e_n)| = \sum_{n=1}^{\infty} |T^*(x^*)(e_n)|,$$

el cual es finito, ya que $\sum_{n=1}^{\infty} e_n$ es WUC. \square

2.2.9 Proposición. *Sea $\sum_{n=1}^{\infty} x_n$ una serie WUC en un espacio de Banach X . Entonces $\sum_{n=1}^{\infty} x_n$ converge incondicionalmente en X si y solo si el operador $T : c_0 \rightarrow X$ tal que $T e_n = x_n$ es compacto.*

Demostración. Supongamos que $\sum_{n=1}^{\infty} x_n$ es incondicionalmente convergente. Vamos a ver que $\lim_{n \rightarrow \infty} \|T - TS_n\| = 0$, donde $(S_n)_{n=1}^{\infty}$ son las proyecciones de las sumas parciales asociadas a la base canónica $(e_n)_{n=1}^{\infty}$ de c_0 . De este modo, por ser el límite uniforme de operadores de rango finito, T será compacto.

Dado $\varepsilon > 0$ usamos el Lema 2.2.3 para encontrar $n = n(\varepsilon)$ tal que si F es un subconjunto finito de $\{n+1, n+2, \dots\}$, entonces $\left\| \sum_{j \in F} x_j \right\| \leq \varepsilon/2$. Para todo $x^* \in X^*$ con $\|x^*\| \leq 1$ tenemos

$$\sum_{\{j \in F : x^*(x_j) \geq 0\}} x^*(x_j) \leq \frac{\varepsilon}{2},$$

y por tanto

$$\sum_{j \in F} |x^*(x_j)| \leq \varepsilon.$$

Por lo tanto si $\xi \in c_{00}$ con $\|\xi\| \leq 1$, entonces $|x^*(T - TS_m)\xi| \leq \varepsilon$ para $m \geq n$ y $x^* \in X^*$. Por densidad concluimos que $\|T - TS_m\| \leq \varepsilon$.

Para la otra implicación asumamos que T es compacto. Consideremos

$$T^{**} : c_0^{**} = \ell_{\infty} \rightarrow X \subset X^{**}.$$

La restricción de T^{**} a $B_{\ell_{\infty}}$ es débil* continua para la norma, porque en la norma de un conjunto compacto, la topología débil* coincide con la topología de la norma. Como $\sum_{n=1}^{\infty} e_{\pi(n)}$ converge débil* en ℓ_{∞} para cada permutación π , $\sum_{n=1}^{\infty} x_n$ también converge incondicionalmente en X . \square

El razonamiento anterior también implica la siguiente propiedad de estabilidad de series incondicionalmente convergentes con respecto a la multiplicación por sucesiones acotadas.

2.2.10 Proposición. *Una serie $\sum_{n=1}^{\infty} x_n$ en un espacio de Banach X es incondicionalmente convergente si y solo si $\sum_{n=1}^{\infty} t_n x_n$ converge (incondicionalmente) para toda $(t_n)_{n=1}^{\infty} \in \ell_{\infty}$.*

Demostración. La implicación trivial es hacia la izquierda. Escogemos como $(t_n)_{n=1}^{\infty}$ la sucesión constante 1 que está en ℓ_{∞} . Obtenemos que $\sum_{n=1}^{\infty} x_n$ converge incondicionalmente.

La implicación restante se razona de la siguiente forma: si $\sum_{n=1}^{\infty} x_n$ converge incondicionalmente entonces por la Proposición 2.2.5 es WUC y por el Lema 2.2.7 existe $C > 0$

tal que para toda $(\xi(n))_{n=1}^{\infty} \in c_{00}$ se cumple

$$\left\| \sum_{n=1}^{\infty} \xi(n) x_n \right\| \leq C \max_n |\xi(n)|.$$

Fijamos $t = (t_n)_{n=1}^{\infty} \in \ell_{\infty}$ y probamos que $\sum_{n=1}^{\infty} t_n x_n$ converge incondicionalmente. Para ello observamos que

$$\left\| \sum_{n=1}^{\infty} \xi(n) t_n x_n \right\| \leq C \|t\|_{\infty} \max_n |\xi(n)|,$$

luego el Lema 2.2.7 nos dice que $\sum_{n=1}^{\infty} t_n x_n$ es WUC.

Ahora, como $\sum_{n=1}^{\infty} x_n$ converge incondicionalmente, la Proposición 2.2.9 implica que el operador $T : c_0 \rightarrow X$ dado por $T e_n = x_n$ es compacto. Como $\sum_{n=1}^{\infty} t_n x_n$ es WUC el operador $S : c_0 \rightarrow X$ dado por $S e_n = t_n x_n$ es continuo.

Si definimos el operador $J \in L(c_0)$ por $J(e_n) = t_n e_n$ para todo $n \in \mathbb{N}$, tenemos que $S(e_n) = t_n T(e_n) = T(t_n e_n) = T(J(e_n))$ para todo $n \in \mathbb{N}$. Luego $S = T \circ J$ y como T es compacto, S también lo es. Finalmente, la Proposición 2.2.9 nos dice que la serie $\sum_{n=1}^{\infty} t_n x_n$ es incondicionalmente convergente en X . \square

2.2.11 Proposición. Una base $(u_n)_{n=1}^{\infty}$ de un espacio de Banach X es incondicional si y solo si existe una constante $K \geq 1$ tal que para todo $N \in \mathbb{N}$,

$$\left\| \sum_{n=1}^N a_n u_n \right\| \leq K \left\| \sum_{n=1}^N b_n u_n \right\| \quad (2.5)$$

siempre que $a_1, \dots, a_N, b_1, \dots, b_N$ sean escalares satisfaciendo $|a_n| \leq |b_n|$ para $n = 1, \dots, N$.

Demostración. Asumimos que $(u_n)_{n=1}^{\infty}$ es incondicional. Si $\sum_{n=1}^N a_n u_n$ converge, entonces también lo hace $\sum_{n=1}^N t_n a_n u_n$ para todo $(t_n)_{n=1}^{\infty} \in \ell_{\infty}$ por la Proposición 2.2.10. Por el teorema de Banach-Steinhaus, la aplicación lineal $T_{(t_n)} : X \rightarrow X$ dada por $\sum_{n=1}^{\infty} a_n u_n \mapsto \sum_{n=1}^{\infty} t_n a_n u_n$ es continua. Ahora el principio de acotación uniforme nos da K de manera que se da la desigualdad.

Demostramos ahora la implicación inversa. Sea $\sum_{n=1}^{\infty} a_n u_n$ una serie convergente en X . Vamos a probar que las subseries $\sum_{k=1}^{\infty} a_{n_k} u_{n_k}$ son convergentes para cada sucesión creciente de enteros $(n_k)_{k=1}^{\infty}$ y apelamos al Lema 2.2.3 para deducir que es incondicionalmente convergente. Dado $\varepsilon > 0$; existe $N = N(\varepsilon) \in \mathbb{N}$ tal que si $m_2 > m_1 \geq N$, entonces

$$\left\| \sum_{n=m_1+1}^{m_2} a_n u_n \right\| < \frac{\varepsilon}{K}.$$

Por hipótesis, si $N \leq n_k < \dots < n_{k+l}$, tenemos

$$\left\| \sum_{j=k+1}^{k+l} a_{n_j} u_{n_j} \right\| \leq K \left\| \sum_{j=n_k+1}^{n_{k+l}} a_j u_j \right\| < \varepsilon,$$

y por tanto $\sum_{k=1}^{\infty} a_{n_k} u_{n_k}$ es de Cauchy. \square

2.2.12 Definición. Sea $(u_n)_{n=1}^{\infty}$ una base incondicional de un espacio de Banach X . La constante básica incondicional K_u de $(u_n)_{n=1}^{\infty}$ es la menor constante K tal que la desigualdad 2.5 es cierta. Entonces decimos que $(u_n)_{n=1}^{\infty}$ es K -incondicional siempre que $K \geq K_u$.

Supongamos que $(u_n)_{n=1}^{\infty}$ es una base incondicional para un espacio de Banach X . Para cada sucesión de escalares $(\alpha_n)_{n=1}^{\infty}$ con $|\alpha_n| = 1$, sea $T_{(\alpha_n)} : X \rightarrow X$ el isomorfismo definido por $T_{(\alpha_n)}(\sum_{n=1}^N a_n u_n) = \sum_{n=1}^N \alpha_n a_n u_n$. Entonces

$$K_u = \sup \{ \|T_{(\alpha_n)}\| : (\alpha_n) \text{ escalares, } |\alpha_n| = 1 \text{ para todo } n \}.$$

Sea $(u_n)_{n=1}^{\infty}$ una base incondicional de X . Para cada $A \subseteq \mathbb{N}$ existe una proyección lineal P_A de X sobre $[u_k : k \in A]$ definida para cada $x = \sum_{k=1}^{\infty} u_k^*(x) u_k$ en X por

$$P_A(x) = \sum_{k \in A} u_k^*(x) u_k.$$

Los miembros del conjunto $\{P_A : A \subseteq \mathbb{N}\}$ son las proyecciones naturales asociadas a la base incondicional $(u_n)_{n=1}^{\infty}$.

2.2.13 Proposición. Sea $(u_n)_{n=1}^{\infty}$ una base de un espacio de Banach X . Los siguientes enunciados son equivalentes:

- (I) La base $(u_n)_{n=1}^{\infty}$ es incondicional.
- (II) La aplicación P_A está bien definida para cada $A \subseteq \mathbb{N}$.
- (III) La aplicación P_A está bien definida para cada $A \subseteq \mathbb{N}$ y $\sup_A \|P_A\| < \infty$.
- (IV) $\sup \{ \|P_F(x)\| : F \subseteq \mathbb{N}, F \text{ finito} \} < \infty$.
- (V) La aplicación P_B está bien definida para cada subconjunto cofinito B de \mathbb{N} y

$$\sup \{ \|P_B(x)\| : B \text{ conjunto cofinito de } \mathbb{N} \} < \infty$$

Además, si alguno de los enunciados de arriba es cierto entonces los supremos de III, IV y V coinciden.

Demostración. La implicación (I) \Rightarrow (II) es una consecuencia de la Proposición 2.2.11.

- (II) \Rightarrow (III) se deduce rápidamente del principio de acotación uniforme.
- (III) \Rightarrow (IV) y (III) \Rightarrow (V) son triviales.
- (IV) \Rightarrow (III) Sea A cualquier subconjunto (finito o infinito) de \mathbb{N} . Para cada $x \in X$ con

soporte finito en $(u_n)_{n=1}^\infty$ sea $S = \text{sop}(x)$. Como $P_A(x) = P_{A \cap S}(x)$,

$$\|P_A(x)\| = \|P_{A \cap S}(x)\| \leq \|P_{A \cap S}\| \|x\| \leq \sup\{\|P_F\| : F \text{ finito}\} \|x\|.$$

Por densidad P_A se extiende a un operador acotado de X en X .

Para cerrar el círculo de equivalencias probaremos (v) \Rightarrow (III). Sea A cualquier subconjunto de \mathbb{N} . Tomamos $x \in X$ con soporte finito en $(u_n)_{n=1}^\infty$ y sea $S = \text{sop}(x)$. Como $P_A(x) = P_{A \cup S^c}(x)$,

$$\|P_A(x)\| = \|P_{A \cup S^c}\| \|x\| \leq \sup\{\|P_B\| : B \subseteq \mathbb{N} \text{ cofinito}\} \|x\|. \quad \square$$

2.2.14 Definición. Si $(u_n)_{n=1}^\infty$ es una base incondicional de X , llamamos al número

$$K_{su} = \sup\{\|P_A\| : A \subseteq \mathbb{N}\}$$

la constante de supresión-incondicional de la base.

Observemos que en general, tenemos

$$1 \leq K_{su} \leq K_u \leq 2K_{su}.$$

2.2.2. Bases simétricas

A continuación estudiamos un tipo especial de bases que incluye a las bases canónicas de los espacios ℓ_p y c_0 .

2.2.15 Definición. Una base $(e_n)_{n=1}^\infty$ de un espacio de Banach X es simétrica si $(e_n)_{n=1}^\infty$ es equivalente a $(e_{\pi(n)})_{n=1}^\infty$ para toda permutación π de \mathbb{N} .

Las bases simétricas son en particular incondicionales. También tienen la propiedad de ser equivalentes a todas sus subsucesiones (infinitas), como dice el próximo lema.

2.2.16 Lema. Supongamos $(e_n)_{n=1}^\infty$ es una base simétrica de un espacio de Banach X . Entonces existe una constante D tal que

$$D^{-1} \left\| \sum_{n=1}^N a_i e_{j_i} \right\| \leq \left\| \sum_{n=1}^N a_i e_{k_i} \right\| \leq D \left\| \sum_{n=1}^N a_i e_{j_i} \right\|$$

para cada $N \in \mathbb{N}$, cada elección de escalares $(a_i)_{i=1}^N$ y cualesquiera dos familias de números naturales $\{j_1, \dots, j_N\}$ y $\{k_1, \dots, k_N\}$.

Demostración. Es suficiente probar el lema para la sucesión básica $(e_n)_{n \geq n_0}$ para algún n_0 . Si es falso, entonces para cada n_0 podemos construir una sucesión estrictamente creciente

de números naturales $(p_n)_{n=0}^{\infty}$ con $p_0 = 0$, números naturales $m_n \leq p_n - p_{n-1}$, escalares $(a_{n,i})_{n=1, i=1}^{\infty, m_n}$, y familias $\{j_{n,1}, \dots, j_{n,m_n}\}$ y $\{k_{n,1}, \dots, k_{n,m_n}\}$ tales que para todo $n = 1, 2, \dots$ tenemos

$$p_{n-1} + 1 \leq j_{n,i}, k_{n,i} \leq p_n, \quad 1 \leq i \leq m_n,$$

$$\left\| \sum_{i=1}^{m_n} a_{n,i} e_{j_{n,i}} \right\| < 2^{-n},$$

y

$$\left\| \sum_{i=1}^{m_n} a_{n,i} e_{k_{n,i}} \right\| > 2^n.$$

Ahora podemos hacer una permutación π de \mathbb{N} tal que $\pi[p_{n-1} + 1, p_n] = [p_{n-1} + 1, p_n]$ y $\pi(j_{n,i}) = k_{n,i}$, y esto contradice la equivalencia de $(e_n)_{n=1}^{\infty}$ y $(e_{\pi(n)})_{n=1}^{\infty}$. \square

2.2.17 Nota. El recíproco del lema no tiene por qué ser cierto. De hecho, la base sumante de c_0 es equivalente a todas sus subsucesiones y no es ni siquiera incondicional.

2.2.18 Definición. Una base $(e_n)_{n=1}^{\infty}$ de un espacio de Banach X es *subsimétrica* si es incondicional y para cada sucesión creciente de enteros $\{n_i\}_{i=1}^{\infty}$, la subbase $(e_{n_i})_{i=1}^{\infty}$ es equivalente a $(e_n)_{n=1}^{\infty}$.

El Lema 2.2.16 dice que las bases simétricas son subsimétricas. Sin embargo, estos dos conceptos no coinciden tal y como muestra el siguiente ejemplo, dado por Garling [5].

2.2.19 Ejemplo. Una base subsimétrica que no es simétrica.

Sea X el espacio de Banach de todas las sucesiones de escalares $\xi = (\xi_n)_{n=1}^{\infty}$ para las cuales

$$\|\xi\| = \sup \sum_{k=1}^{\infty} \frac{|\xi_{n_k}|}{\sqrt{k}} < \infty,$$

se toma el supremo de todas las sucesiones crecientes de enteros $(n_k)_{k=1}^{\infty}$. X dotado de esta norma es un espacio de Banach cuyos vectores unidad $(e_n)_{n=1}^{\infty}$ forman una base subsimétrica que no es simétrica.

Sea $(e_n)_{n=1}^{\infty}$ una base simétrica de un espacio de Banach X . Para cada permutación π de \mathbb{N} y cada sucesión de signos $\varepsilon = (\varepsilon_n)_{n=1}^{\infty}$, existe un automorfismo

$$T_{\pi, \varepsilon} : X \rightarrow X, \quad x = \sum_{n=1}^{\infty} a_n e_n \mapsto T_{\pi, \varepsilon}(x) = \sum_{n=1}^{\infty} \varepsilon_n a_{\pi(n)} e_{\pi(n)}.$$

El principio de acotación uniforme nos da un número K tal que

$$\sup_{\pi, \varepsilon} \|T_{\pi, \varepsilon}\| \leq K,$$

i.e., la estimación

$$\left\| \sum_{n=1}^{\infty} \varepsilon_n a_n e_{\pi(n)} \right\| \leq K \left\| \sum_{n=1}^{\infty} a_n e_n \right\| \quad (2.6)$$

es válida para todas las elecciones de signos $(\varepsilon_n)_{n=1}^{\infty}$ y todas las permutaciones π .

La constante más pequeña $1 \leq K$ en 2.6 es llamada la *constante simétrica* de $(e_n)_{n=1}^{\infty}$ y se denotará por K_s . Entonces diremos que $(e_n)_{n=1}^{\infty}$ es K -simétrica siempre que $K_s \leq K$.

Para cada $x = \sum_{n=1}^{\infty} a_n e_n \in X$, pongamos

$$\|x\| = \sup \left\| \sum_{n=1}^{\infty} \varepsilon_n a_n e_{\pi(n)} \right\|, \quad (2.7)$$

el supremo tomado sobre todas las opciones de escalares $(\varepsilon_n)_{n=1}^{\infty}$ de signos y todas las permutaciones de números naturales. La ecuación 2.7 define una nueva norma en X equivalente a $\|\cdot\|$, ya que $\|x\| \leq \|x\| \leq K \|x\|$ para todo $x \in X$. Con respecto a esta norma, $(e_n)_{n=1}^{\infty}$ es una base 1-simétrica de X .

2.2.3. Bases y dualidad: bases acotadamente completas y bases shrinking

Supongamos que $(e_n)_{n=1}^{\infty}$ es una base de un espacio de Banach X y que $(e_n^*)_{n=1}^{\infty}$ es la sucesión de funcionales biortogonales. Uno de los objetivos de este subapartado es establecer condiciones necesarias y suficientes para que $(e_n^*)_{n=1}^{\infty}$ sea una base de X^* . Este no es siempre el caso, ya que es necesario que X^* sea un espacio separable. De hecho, existen espacios de Banach X con una base y con dual separable tal que X^* no tiene la propiedad de aproximación; por lo tanto, X^* no puede tener una base. Sin embargo $(e_n^*)_{n=1}^{\infty}$ es siempre una sucesión básica en X^* .

A continuación introducimos un lema que necesitaremos en varias demostraciones.

2.2.20 Lema. (*Vuelta parcial del teorema de Banach-Steinhaus*). Sea $(T_n)_{n=1}^{\infty}$ una sucesión de operadores lineales continuos de un espacio de Banach X en un espacio lineal normado Y tal que $\sup_n \|T_n\| < \infty$. Si $T : X \rightarrow Y$ es otro operador, entonces el subespacio

$$\{x \in X : \|T_n(x) - T(x)\| \rightarrow 0\}$$

es cerrado para la norma en X .

2.2.21 Proposición. Sea $(e_n)_{n=1}^{\infty}$ una base para un espacio de Banach X con constante básica K_b y funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$. Entonces $(e_n^*)_{n=1}^{\infty}$ es una base para el subespacio $Z = [e_n^*]$ con constante básica no mayor a K_b . Más específicamente:

- (a) Los funcionales coordenados $(e_n^{**})_{n=1}^{\infty}$ asociados a los funcionales $(e_n^*)_{n=1}^{\infty}$ están dados por $e_n^{**} = j(e_n)|_Z$ para cada n , donde j es el embebimiento canónico de X en su dual segundo X^{**} .
- (b) Las proyecciones de las sumas parciales asociadas a la sucesión básica $(e_n)_{n=1}^{\infty}$ son las restricciones $(S_N^*|_Z)_{N=1}^{\infty}$ de los operadores adjuntos de las proyecciones de las sumas parciales asociadas a $(e_n)_{n=1}^{\infty}$. Más aún,

$$Z = \left\{ x^* \in X^* : \lim_{N \rightarrow \infty} S_N^*(x^*) = x^* \right\}$$

Demostración. Los funcionales $(e_n^{**})_{n=1}^{\infty}$ satisfacen $e_n^{**}(e_k^*) = 1$ si $n = k$ y 0 en otro caso. Basta probar que los operadores $T_N : Z \rightarrow Z$ definidos por

$$T_N(x^*) = \sum_{k=1}^N e_k^{**}(x^*) e_k^*, \quad x^* \in Z,$$

satisfacen $\|T_N\| \leq K_b$ y usar la vuelta parcial del teorema de Banach-Steinhaus.

Sea $(S_N^*)_{N=1}^{\infty}$ la sucesión de operadores adjuntos de las proyecciones de las sumas parciales asociadas a $(e_n)_{n=1}^{\infty}$. Para cada $x \in X$ y $x^* \in X^*$:

$$S_N^*(x^*)(x) = x^*(S_N(x)) = \sum_{k=1}^N e_k^*(x)x^*(e_k) = \sum_{k=1}^N e_k^*(x)j(e_k)(x^*).$$

Por tanto $S_N^* = \sum_{k=1}^N j(e_k)e_k^*$ y $T_N = S_N^*|_Z$. De este modo:

$$\sup_N \|T_N\| \leq \sup_N \|S_N^*\| = \sup_N \|S_N\| = K_b.$$

□

2.2.22 Ejemplo. La identificación natural de ℓ_1^* con ℓ_∞ muestra que $(e_n^*)_{n=1}^{\infty}$ es una base para el subespacio de ℓ_1^* isométricamente isomorfo a c_0 .

Vamos a introducir una definición que nos será útil en la siguiente proposición.

2.2.23 Definición. Supongamos que X es un espacio normado e Y es un subespacio de X^* . Consideremos una nueva norma en X dada por

$$\|x\|_Y = \sup\{|y^*(x)| : y^* \in Y, \|y^*\| = 1\}, \quad x \in X,$$

Si existe una constante $c \leq 1$ tal que

$$c\|x\| \leq \|x\|_Y \leq \|x\|, \quad x \in X,$$

entonces se dice que Y es un subespacio c -normante para X en X^* .

En términos generales, la siguiente proposición dice que si $(e_n)_{n=1}^{\infty}$ es una base para un espacio de Banach X , entonces el subespacio $Z = [e_n^*]$ de X^* es razonablemente grande, en el sentido de que es $1/K_b$ -normante para X .

2.2.24 Proposición. *Sea $(e_n)_{n=1}^{\infty}$ una base de un espacio de Banach X con constante básica K_b y sea $Z = [e_n^*]$. La norma en X definida por*

$$\|x\|_Z = \sup\{|h(x)| : h \in Z, \|h\| \leq 1\}$$

satisface

$$K_b^{-1} \|x\| \leq \|x\|_Z \leq \|x\|, \quad x \in X.$$

Así:

- (i) *El subespacio Z de X^* es $1/K_b$ -normante para X .*
- (ii) *La aplicación $x \rightarrow j(x)|_Z$ define un embebimiento isomórfico de X en Z^* . Este embebimiento es isométrico si $(e_n)_{n=1}^{\infty}$ es monótona.*

Demostración. Sea $x \in X$. Como $Z \subseteq X^*$,

$$\|x\|_Z \leq \sup\{|x^*(x)| : x^* \in X^*, \|x^*\| \leq 1\} = \|x\|.$$

Para probar la desigualdad contraria cogemos $x^* \in S_{X^*}$ tal que $x^*(x) = \|x\|$. Para cada N ,

$$\frac{|(S_N^* x^*)x|}{K_b} \leq \frac{|(S_N^* x^*)x|}{\|S_N^* x^*\|} \leq \sup\{|h(x)| : h \in Z, \|h\| \leq 1\} = \|x\|_Z.$$

Dejamos que $N \rightarrow \infty$ y usamos que si $\|S_N(x) - x\| \rightarrow 0$ entonces $|S_N^* x^*(x)| = |x^*(S_N x)| \rightarrow \|x\|$. \square

Como consecuencia del embebimiento isomórfico de X en Z^* y la Proposición 2.2.21 obtenemos una propiedad reflexiva interesante para sucesiones básicas.

2.2.25 Corolario. *Sea $(e_n)_{n=1}^{\infty}$ una base de un espacio de Banach X . La sucesión básica $(e_n^{**})_{n=1}^{\infty}$ es equivalente a $(e_n)_{n=1}^{\infty}$ (isométricamente equivalente si $(e_n)_{n=1}^{\infty}$ es monótona).*

Aunque $(e_n^*)_{n=1}^{\infty}$ puede no ser una base para X^* , uno podría ir más lejos y, basado en la siguiente proposición, decir que $(e_n^*)_{n=1}^{\infty}$ es una base débil* para X^* .

2.2.26 Proposición. *Sea $(e_n)_{n=1}^{\infty}$ una base para un espacio de Banach X con funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$. Entonces para cada $x^* \in X^*$ existe una única sucesión de escalares $(a_n)_{n=1}^{\infty}$ tal que*

$$x^* = \sum_{n=1}^{\infty} a_n e_n^*,$$

la serie converge en la topología débil* de X^* . Más precisamente,

$$x^* = \text{débil}^* - \lim_{N \rightarrow \infty} \sum_{n=1}^N x^*(e_n) e_n^*.$$

Demostración. Para cada $x \in X$,

$$|(x^* - S_N^*(x^*))(x)| = |(x^*(x - S_N(x)))| \leq \|x^*\| \|x - S_N(x)\| \xrightarrow{N \rightarrow \infty} 0. \quad \square$$

De la Proposición 2.2.21 (1) deducimos que $(e_n^*)_{n=1}^\infty$ será una base para X^* (en el sentido habitual) si y solo si $[e_n^*] = X^*$. Nuestro próximo resultado proporciona una comprobación útil para esto, pero primero veamos un caso trivial.

2.2.27 Proposición. Si $(e_n)_{n=1}^\infty$ es una base para un espacio de Banach reflexivo X , entonces $(e_n^*)_{n=1}^\infty$ es una base para X^* .

Demostración. La Proposición 2.2.26 nos dice que el cierre lineal de $(e_n^*)_{n=1}^\infty$ es denso débil* en X^* . Como X es reflexivo, las topologías débil y la débil* de X^* coinciden, por lo que el cierre lineal de $(e_n^*)_{n=1}^\infty$ es de hecho denso débilmente en X^* . Una aplicación del teorema de Mazur (expuesto a continuación) asegura que el cierre lineal de $(e_n^*)_{n=1}^\infty$ es denso para la norma en X^* , que junto a la Proposición 2.2.21 (a) da que $(e_n^*)_{n=1}^\infty$ es una base para X^* . \square

2.2.28 Teorema. (Teorema de Mazur). Si C es un conjunto convexo en un espacio normado X , entonces la clausura de C en la norma de la topología, \bar{C} , coincide con \bar{C}^w , la clausura de C en la topología débil.

2.2.29 Ejemplo. Supongamos $1 < p < \infty$. Si $(e_n)_{n=1}^\infty$ es una base para ℓ_p [respectivamente, $L_p[0, 1]$], entonces $(e_n^*)_{n=1}^\infty$ es una base para ℓ_q [respectivamente $L_q[0, 1]$], donde $\frac{1}{p} + \frac{1}{q} = 1$.

2.2.30 Proposición. Supongamos que $(e_n)_{n=1}^\infty$ es una base para un espacio de Banach X . Los funcionales coordenados $(e_n^*)_{n=1}^\infty$ son una base para X^* si y solo si

$$\lim_{N \rightarrow \infty} \|x^*\|_N = 0 \text{ para cada } x^* \in X^*, \quad (2.8)$$

donde $\|x^*\|_N$ es la norma de x^* restringida al espacio (cola) $[e_n]_{n>N}$, i.e.,

$$\|x^*\|_N = \sup \{|x^*(y)| : y \in [e_n]_{n>N}, \|y\| \leq 1\}.$$

Demostración. Supongamos $(e_n^*)_{n=1}^\infty$ es una base para X^* . Cada $x^* \in X^*$ puede descomponerse como $(x^* - S_N^* x^*) + S_N^* x^*$ para cada N . Entonces

$$\|x^*\|_N \leq \|(x^* - S_N^* x^*)|_{[e_n]_{n>N}}\| + \|S_N^* x^*|_{[e_n]_{n>N}}\| \leq \|x^* - S_N^* x^*\| \xrightarrow{N \rightarrow \infty} 0.$$

Asumamos ahora que 2.8 es válido. Sea x^* un elemento de X^* . Como para cada $x \in X$, $(I_X - S_N)(x)$ está en el subespacio $[e_n]_{n>N}$, tenemos que

$$\begin{aligned} |(x^* - S_N^* x^*)(x)| &= |x^*(I_X - S_N)(x)| \\ &\leq \|x^*\|_{[e_n]_{n \geq N+1}} \|I_X - S_N\| \|x\| \\ &\leq (K_b + 1) \|x^*\|_{[e_n]_{n \geq N+1}} \|x\|. \end{aligned}$$

Por tanto

$$\|x^* - S_N^* x^*\| \leq (K_b + 1) \|x^*\|_{[e_n]_{n \geq N+1}} \xrightarrow{N \rightarrow \infty} 0.$$

Así $X^* = [e_n^*]$ y hemos acabado. \square

2.2.31 Definición. Decimos que una base $(e_n)_{n=1}^\infty$ que cumpla la propiedad 2.8 es una base shrinking.

2.2.32 Ejemplo. La base unitaria de vectores $(e_n)_{n=1}^\infty$ de c_0 es shrinking, ya que $c_0^* = \ell_1$ y dado $x^* = (a_n)_{n=1}^\infty \in \ell_1$ tenemos

$$\|x^*\|_N = \sum_{n=N+1}^{\infty} |a_n| \xrightarrow{N \rightarrow \infty} 0.$$

En general, solo porque un espacio tenga una base shrinking no significa que toda base para ese espacio sea shrinking también. La base sumante $(f_n)_{n=1}^\infty$ de c_0 , por ejemplo, no es shrinking. Para ver esto podemos tomar el funcional coordenado e_1^* correspondiente a el primer vector de la base canónica de c_0 y simplemente observar que $e_1^*(f_n) = 1$ para todo n , por lo que la condición 2.8 no se puede dar.

A continuación damos una caracterización adicional de bases shrinking.

2.2.33 Proposición. Una base $(e_n)_{n=1}^\infty$ de un espacio de Banach X es shrinking si y solo si cada sucesión básica por bloques acotada de $(e_n)_{n=1}^\infty$ es débilmente nula.

Demostración. Asumimos que $(e_n)_{n=1}^\infty$ no es shrinking. Entonces $Z \neq X^*$; por lo tanto existe $x^* \in X^* \setminus [e_n^*]$, $\|x^*\| = 1$, tal que por la Proposición 2.2.26 la serie $\sum_{n=1}^\infty x^*(e_n)e_n^*$ converge a x^* en la topología débil* de X^* pero no converge en la topología de la norma de X^* . Usando la condición de Cauchy, podemos encontrar dos sucesiones de naturales positivos $(p_n)_{n=1}^\infty$, $(q_n)_{n=1}^\infty$ y $\delta > 0$ tal que $p_1 \leq q_1 < p_2 \leq q_2 < p_3 \leq q_3 < \dots$ y $\left\| \sum_{n=p_k}^{q_k} x^*(e_n)e_n^* \right\| > \delta$ para todo $k \in \mathbb{N}$. Así para cada k existe $x_k \in X$, $\|x_k\| = 1$, para el cual $\left\| \sum_{n=p_k}^{q_k} x^*(e_n)e_n^*(x_k) \right\| > \delta$. Sea

$$y_k = \sum_{n=p_k}^{q_k} e_n^*(x_k)e_n^*, \quad k = 1, 2, \dots$$

La sucesión $(y_k)_{k=1}^{\infty}$ es una sucesión básica por bloques de $(e_n)_{n=1}^{\infty}$ que no es débilmente nula, ya que $x^*(y_k) > \delta$ para todo k .

La implicación inversa se deduce inmediatamente de la Proposición 2.2.30. \square

Una definición compañera a la de bases shrinking que fue dada por James como una herramienta para el estudio de la estructura de los espacios de Banach es la de bases *acotadamente completas*. Esta propiedad había sido usada por Dunford y Morse en 1936 para garantizar la existencia de derivadas de aplicaciones de Lipschitz en la recta real tomando valores en espacios de Banach.

2.2.34 Definición. Sea X un espacio de Banach. Una base $(e_n)_{n=1}^{\infty}$ de X es una base acotadamente completa si para cualquier sucesión de escalares $(a_n)_{n=1}^{\infty}$ tal que

$$\sup_N \left\| \sum_{n=1}^N a_n e_n \right\| < \infty,$$

entonces la serie $\sum_{n=1}^{\infty} a_n e_n$ converge.

2.2.35 Ejemplo. (a) La base canónica de ℓ_p para $1 \leq p < \infty$ es acotadamente completa.

(b) La base natural $(e_n)_{n=1}^{\infty}$ de c_0 no es acotadamente completa. En efecto, la serie $\sum_{n=1}^{\infty} e_n$ no es convergente en c_0 a pesar de que

$$\sup_N \left\| \sum_{n=1}^N e_n \right\|_{\infty} = \sup_N \left\| (\underbrace{1, 1, \dots, 1}_N, 0, 0, \dots) \right\|_{\infty} = 1.$$

(c) La base sumante $(f_n)_{n=1}^{\infty}$ de c_0 no es acotadamente completa, ya que

$$\sup_N \left\| \sum_{n=1}^N (-1)^n f_n \right\|_{\infty} = 1,$$

pero la serie $\sum_{n=1}^{\infty} (-1)^n f_n$ no converge.

Los dos próximos teoremas mostrarán que las bases acotadamente completas y las bases shrinking están en dualidad. Antes de enunciar estos resultados y probarlos deberíamos enunciar un lema sobre bases acotadamente completas que es de interés y destaca una propiedad que necesitamos.

2.2.36 Lema. Supongamos $(e_n)_{n=1}^{\infty}$ es una base acotadamente completa para un espacio de Banach X con funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$. Entonces para cada $x^{**} \in X^{**}$ tenemos

$$\lim_{N \rightarrow \infty} \sum_{n=1}^N x^{**}(e_n^*) e_n \in X.$$

Demostración. Para cada $N \in \mathbb{N}$,

$$\sum_{n=1}^N x^{**}(e_n^*) e_n = S_N^{**}(x^{**}),$$

donde S_N^{**} es el dual doble de S_N . Por lo tanto,

$$\left\| \sum_{n=1}^N x^{**}(e_n^*) e_n \right\| = \|S_N^{**}(x^{**})\| \leq \sup_N \|S_N^{**}\| \|x^{**}\| = K_b \|x^{**}\|.$$

Por ser $(e_n)_{n=1}^\infty$ acotadamente completa tenemos $\lim_N \sum_{n=1}^N x^{**}(e_n^*) e_n \in X$. \square

Vamos a recordar un teorema que nos será útil más adelante.

2.2.37 Teorema. (*Teorema de Banach-Alaoglu*). Si X es un espacio lineal normado, entonces el conjunto $B_{X^*} = \{x^* \in X^* : \|x^*\| \leq 1\}$ es debil* compacto.

2.2.38 Teorema. Sea $(e_n)_{n=1}^\infty$ una base para un espacio de Banach X con funcionales biortogonales $(e_n^*)_{n=1}^\infty$. Son equivalentes:

- (I) $(e_n)_{n=1}^\infty$ es una base acotadamente completa para X .
- (II) $(e_n^*)_{n=1}^\infty$ es una base shrinking para $Z = [e_n^*]$, i.e., la sucesión de sus funcionales biortogonales $(j(e_n)|_Z)_{n=1}^\infty$ es una base para Z^* .
- (III) La aplicación $x \mapsto j(x)|_Z$ define un isomorfismo de X en Z^* que es isométrico si $K_b = 1$.

Demostración. (I) \Rightarrow (III) Usando la Proposición 2.2.24 solo necesitamos ver que la aplicación es sobreyectiva. Dado $h^* \in Z^*$, existen $x^{**} \in X^{**}$ tal que $x^{**}|_Z = h^*$. Por el Lema 2.2.36 la serie $\sum_{n=1}^\infty x^{**}(e_n^*) e_n$ converge a algún $x \in X$. Ahora $j(x)|_Z = h^*$, ya que para cada $k \in \mathbb{N}$ tenemos

$$j(x)|_Z(e_k^*) = e_k^*(x) = x^{**}(e_k^*) = h^*(e_k^*).$$

(III) \Rightarrow (II) Asumimos que $x \mapsto j(x)|_Z$ es un isomorfismo de X en Z^* . Entonces $(j(e_n)|_Z)_{n=1}^\infty$ es una base para Z^* y es también la sucesión de funcionales coordenados para $(e_n^*)_{n=1}^\infty$. Esto significa que $(e_n^*)_{n=1}^\infty$ es una base shrinking para Z .

(II) \Rightarrow (I) Sea $(a_n)_{n=1}^\infty$ una sucesión de escalares para los cuales

$$\sup_N \left\| \sum_{n=1}^N a_n e_n \right\| < \infty.$$

Como $(\sum_{n=1}^N a_n j(e_n))_{N=1}^\infty$ está acotada en X^{**} , entonces por el teorema de Banach-Alaoglu (2.2.37) existe un punto de acumulación debil* $x^{**} \in X^{**}$ de esa sucesión. Sea $h^* = x^{**}|_Z$.

Como

$$\lim_N \sum_{n=1}^N a_n j(e_n)(e_k^*) = \lim_N \sum_{n=1}^N a_n e_k^*(e_n) = a_k,$$

tenemos que $h^*(e_k^*) = x^{**}(e_k^*) = a_k$ para todo k . Usando la hipótesis y la Proposición 2.2.21, obtenemos

$$h^* = \sum_{n=1}^{\infty} e_n^{***}(h^*) e_n^{**} = \sum_{n=1}^{\infty} h^*(e_n^*) e_n^{**} = \sum_{n=1}^{\infty} a_n j(e_n)|_Z.$$

En particular, la serie anterior converge en Z^* . Como por la Proposición 2.2.24, la aplicación $x \mapsto j(x)|_Z$ es un embebimiento isomórfico, la serie $\sum_{n=1}^{\infty} a_n e_n$ converge en X . \square

2.2.39 Corolario. *El espacio c_0 no tiene una base acotadamente completa.*

Demostración. El resultado se sigue del Teorema 2.2.38, teniendo en cuenta que c_0 no es isomórfico a un espacio dual. \square

2.2.40 Teorema. *Sea $(e_n)_{n=1}^{\infty}$ una base para un espacio de Banach X con funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$. Son equivalentes:*

- (I) $(e_n)_{n=1}^{\infty}$ es una base shrinking para X .
- (II) $(e_n^*)_{n=1}^{\infty}$ es una base acotadamente completa para $Z = [e_n^*]$.
- (III) $Z = X^*$.

Demostración. Simplemente aplicar el Teorema 2.2.38 a la base $(e_n^*)_{n=1}^{\infty}$ de Z teniendo en cuenta el Corolario 2.2.25. \square

En 1948, Karlin planteó la siguiente pregunta: Si el dual X^* de un espacio de Banach X tiene una base, ¿tiene X una base? Johnson, Rosenthal, y Zippin dieron en 1971 una respuesta afirmativa a esta pregunta, resolviendo uno de los problemas fundamentales en la teoría de bases. Ellos mostraron que de hecho X tiene una base shrinking. De su resultado se sigue que si X tiene una base y X^* es separable y satisface la propiedad de aproximación, entonces X^* tiene una base. Una pregunta relacionada pero más fácil es esta: Si X tiene una base $(e_n)_{n=1}^{\infty}$, ¿qué propiedad de $(e_n)_{n=1}^{\infty}$ nos dice si X es el dual de otro espacio de Banach con una base? El Teorema 2.2.38 nos da la respuesta: si $(e_n)_{n=1}^{\infty}$ es una base acotadamente completa, entonces X es isomorfo al espacio dual $Z^* = [e_n^*]^*$.

Puede que no sepamos mucho más de X^* más allá de que tiene una base acotadamente completa si todo lo que sabemos de X es que tiene una base shrinking. Sin embargo, es posible dar una descripción muy útil de X^{**} como el espacio de todas las sucesiones de escalares $(a_n)_{n=1}^{\infty}$ para las cuales $\sup_N \sum_{n=1}^N a_n e_n < \infty$. El propósito original del siguiente

teorema era sentar las bases del espacio de James \mathcal{J} . Su prueba usa una combinación de ideas esencialmente contenidas en la demostración del Teorema 2.2.38 por lo que la vamos a omitir.

2.2.41 Teorema. *Supongamos que $(e_n)_{n=1}^{\infty}$ es una base shrinking para un espacio de Banach X con funcionales biortogonales $(e_n^*)_{n=1}^{\infty}$. La aplicación*

$$x^{**} \mapsto (x^{**}(e_n^*))_{n=1}^{\infty}$$

define un isomorfismo de X^{**} con el espacio de todas las sucesiones de escalares $(a_n)_{n=1}^{\infty}$ tal que $\sup_N \left\| \sum_{n=1}^N a_n e_n \right\| < \infty$. En particular,

$$\|x^{**}\| \approx \sup_N \left\| \sum_{n=1}^N x^{**}(e_n^*) e_n \right\|,$$

y si $(e_n)_{n=1}^{\infty}$ es monótona, entonces

$$\|x^{**}\| \approx \lim_{N \rightarrow \infty} \left\| \sum_{n=1}^N x^{**}(e_n^*) e_n \right\|.$$

La imagen canónica de X dentro de X^{**} corresponde a todas las sucesiones de escalares $(a_n)_{n=1}^{\infty}$ para las cuales $(\sum_{n=1}^N a_n e_n)_{n=1}^{\infty}$ no sólo está acotada sino que converge en norma.

Ahora vamos a exponer el resultado principal de este subapartado, que se debe a James [6].

2.2.42 Teorema. *Sea X un espacio de Banach. Si X tiene una base $(e_n)_{n=1}^{\infty}$ entonces X es reflexivo si y solo si $(e_n)_{n=1}^{\infty}$ es una base acotadamente completa y shrinking.*

Demostración. Asumimos que X es reflexivo y que $(e_n)_{n=1}^{\infty}$ es una base para X . Entonces $X^* = Z$. Si no, usando el teorema de Hahn-Banach, uno puede encontrar $0 \neq x^{**} \in X^{**}$ tal que $x^{**}(h) = 0$ para todo $h \in Z$. Por reflexividad existe $0 \neq x = \sum_{n=1}^{\infty} e_n^*(x) e_n \in X$ tal que $x = x^{**}$. En particular, tendríamos $0 = x^{**}(e_n^*) = e_n^*(x)$ para todo n , lo cual implicaría $x = 0$. Así $(e_n)_{n=1}^{\infty}$ es shrinking. Ahora consideremos $(e_n^*)_{n=1}^{\infty}$ como una base para el espacio de Banach reflexivo $X^* = Z$. Lo anterior muestra que $(e_n^*)_{n=1}^{\infty}$ es shrinking; por tanto por el Teorema 2.2.38, $(e_n)_{n=1}^{\infty}$ es acotadamente completa. \square

Este teorema da un criterio para la reflexividad que es muy útil, particularmente en la construcción de ejemplos. El hecho de que la base canónica de ℓ_1 no sea shrinking y que la base canónica de c_0 no sea acotadamente completa se explican ahora por la no reflexividad de estos espacios.

Durante la década de 1960 estaba de moda estudiar la estructura de los espacios de Banach entendiendo las propiedades de sus bases. Por supuesto, este hecho se vio algo

debilitado cuando Enflo demostró que no todos los espacios de Banach separables tienen una base. Uno de los puntos álgidos de esta teoría fue el teorema de Zippin el cual dice que un espacio de Banach con una base es reflexivo si y solo si toda base está acotadamente completa o si y solo si toda base es shrinking. Así, todo espacio de Banach no reflexivo que tiene una base debe de tener al menos una base que no sea acotadamente completa y al menos una base que no sea shrinking.

2.3. La base de Haar en $L_p[0, 1]$ ($1 \leq p < \infty$)

En esta sección presentaremos la base de Haar en $L_p[0, 1]$ ($1 \leq p < \infty$) y estudiaremos algunas de sus propiedades. Continuaremos el estudio de las base de Haar en la sección 2.4.

El *sistema de Haar* es una sucesión de funciones $(h_n)_{n=1}^{\infty}$ definidas en $[0, 1]$ por $h_1 = 1$ y para $n = 2^k + s$, donde $k = 0, 1, 2, \dots$, y $s = 1, 2, \dots, 2^k$,

$$h_n(t) = \chi_{\left[\frac{2s-2}{2^{k+1}}, \frac{2s-1}{2^{k+1}}\right)}(t) - \chi_{\left[\frac{2s-1}{2^{k+1}}, \frac{2s}{2^{k+1}}\right)}(t) = \begin{cases} 1 & \text{si } \frac{2s-2}{2^{k+1}} \leq t < \frac{2s-1}{2^{k+1}} \\ -1 & \text{si } \frac{2s-1}{2^{k+1}} \leq t < \frac{2s}{2^{k+1}} \\ 0 & \text{en otro caso.} \end{cases}$$

Dados $k = 0, 1, 2, \dots$ y $1 \leq s \leq 2^k$, cada intervalo de la forma $\left[\frac{s-1}{2^k}, \frac{s}{2^k}\right)$ es llamado *diádico*. Es a menudo útil etiquetar los elementos del sistema de Haar por sus soportes; por tanto escribimos h_I para denotar h_n cuando I es el soporte del intervalo diádico de h_n .

En este apartado veremos que el sistema de Haar es una base de Schauder (no normalizada) en L_p para cada $1 \leq p < \infty$ y que es incondicional cuando $1 < p < \infty$. Necesitaremos el concepto de expectativa condicional, que introducimos a continuación.

Sea (Ω, Σ, μ) un espacio de probabilidad, y Σ' una sub- σ -álgebra de Σ . Dada $f \in L_1(\Omega, \Sigma, \mu)$, podemos definir una medida, ν , en Σ' como

$$\nu(E) = \int_E f d\mu, \quad E \in \Sigma'.$$

La medida ν es absolutamente continua con respecto a $\mu|_{\Sigma'}$; por el teorema de Radon-Nikodym, existe una (única, salvo conjuntos de medida cero) función Σ' -medible $\psi \in L_1(\Omega, \Sigma', \mu)$ tal que

$$\nu(E) = \int_E \psi d\mu, \quad E \in \Sigma'$$

2.3.1 Definición. Dada $f \in L_1(\Omega, \Sigma, \mu)$, la expectativa condicional de f en la σ -álgebra Σ' es la (única) función ψ que satisface

$$\int_E f d\mu = \int_E \psi d\mu, \quad \forall E \in \Sigma'.$$

Denotaremos a la función ψ por $\mathbb{E}(f|\Sigma')$.

Tengamos en cuenta que si Σ' consiste en innumerables $(A_n)_{n=1}^{\infty}$ disjuntos, la definición de $\mathbb{E}(f|\Sigma')$ es especialmente simple:

$$\mathbb{E}(f|\Sigma')(t) = \sum_{j=1}^{\infty} \frac{1}{\mu(A_j)} \left(\int_{A_j} f d\mu \right) \chi_{A_j}(t).$$

También observamos que si $f \in L_1(\mu)$, para toda función simple Σ' -medible g , tenemos

$$\int_{\Omega} g f d\mu = \int_{\Omega} g \mathbb{E}(f|\Sigma') d\mu$$

y

$$\mathbb{E}(fg|\Sigma') = g \mathbb{E}(f|\Sigma').$$

2.3.2 Lema. Sea (Ω, Σ, μ) un espacio de probabilidad y supongamos que Σ' es una sub- σ -álgebra de Σ . Entonces $\mathbb{E}(\cdot|\Sigma')$ es una proyección lineal de norma uno de $L_p(\Omega, \Sigma, \mu)$ en $L_p(\Omega, \Sigma', \mu)$ para cada $1 \leq p < \infty$.

Demostración. Fijamos $1 \leq p < \infty$. Es inmediato comprobar que $\mathbb{E}(\cdot|\Sigma')^2 = \mathbb{E}(\cdot|\Sigma')$. Si $f \in L_p(\mu)$, usando la desigualdad de Hölder en $L_p(\Omega, \Sigma', \mu)$, tenemos

$$\begin{aligned} \|\mathbb{E}(\cdot|\Sigma')\|_p &= \sup \left\{ \int_{\Omega} \mathbb{E}(\cdot|\Sigma') g d\mu : g \text{ simple } \Sigma'\text{-medible con } \|g\|_q \leq 1 \right\} \\ &= \sup \left\{ \int_{\Omega} f g d\mu : g \text{ simple } \Sigma'\text{-medible con } \|g\|_q \leq 1 \right\} \\ &\leq \sup \left\{ \int_{\Omega} f g d\mu : g \text{ simple con } \|g\|_q \leq 1 \right\} = \|f\|_p. \end{aligned} \quad \square$$

2.3.3 Proposición. El sistema de Haar es una base monótona en L_p para $1 \leq p < \infty$.

Demostración. Consideremos una sucesión creciente de σ -álgebras, $(\mathcal{B})_{n=1}^{\infty}$, contenida en la σ -álgebra de Borel de $[0, 1]$ definida como sigue: sea \mathcal{B}_1 la σ -álgebra trivial, $\{\emptyset, [0, 1]\}$, y para $n = 2^k + s$ ($k = 0, 1, 2, \dots$, $1 \leq s \leq 2^k$) sea \mathcal{B}_n el subálgebra finito de los conjuntos de Borel de $[0, 1]$ cuyos elementos son los intervalos diádicos de la familia

$$\mathcal{F}_n = \begin{cases} \left[\frac{j-1}{2^{k+1}}, \frac{j}{2^{k+1}} \right) & \text{para } j = 1, \dots, 2s \\ \left[\frac{j-1}{2^k}, \frac{j}{2^k} \right) & \text{para } j = s+1, \dots, 2^k \end{cases}$$

Fijamos $1 \leq p < \infty$. Para cada n , \mathbb{E}_n denotará al operador de expectativa condicional en la σ -álgebra \mathcal{B}_n . Por el Lema 2.3.2 \mathbb{E}_n es una proyección de norma uno de L_p en $L_p([0, 1], \mathcal{B}_n, \lambda)$, el espacio de funciones que son constantes en intervalos de la familia \mathcal{F}_n . Denotaremos este espacio por $L_p(\mathcal{B}_n)$. Claramente, *rango* $\mathbb{E}_n = n$. Además, $\mathbb{E}_n \mathbb{E}_m = \mathbb{E}_m \mathbb{E}_n = \mathbb{E}_{\min\{m,n\}}$ para cualesquiera dos enteros positivos m, n .

Por otro lado, el conjunto

$$\{f \in L_p : \|\mathbb{E}_n(f) - f\| \rightarrow 0\}$$

es cerrado por la vuelta parcial del teorema de Banach-Steinhaus y contiene al conjunto $\bigcup_{k=1}^{\infty} L_p(\mathcal{B}_k)$, el cual es denso en L_p . Por lo tanto $\|\mathbb{E}_n(f) - f\| \rightarrow 0$ para todo $f \in L_p$. Por la Proposición 2.1.7 L_p tiene una base cuya proyecciones naturales son $(\mathbb{E}_n)_{n=1}^{\infty}$. Esta base es de hecho el sistema de Haar, porque para cada $n \in \mathbb{N}$ tenemos $\mathbb{E}_m(h_n) = h_n$ para $m \geq n$ y $\mathbb{E}_m h_n = 0$ para $m < n$. La constante básica es $\sup_n \|\mathbb{E}_n\| = 1$. \square

2.3.4 Notas. (a) El sistema de Haar tal y como lo hemos definido no está normalizado en L_p para $1 \leq p < \infty$. Está normalizado en L_{∞} , ya que $\|h_{2^k+s}\|_p = (1/2^k)^{1/p}$. Para normalizarlo en L_p uno debe tomar $h_n/\|h_n\|_p = |I_n|^{-1/p} h_n$, donde I_n denota el soporte de la función de Haar h_n .

(b) Observemos que si $f \in L_p$ ($1 \leq p < \infty$), entonces

$$\mathbb{E}_n(f) - \mathbb{E}_{n-1}(f) = \left(\frac{1}{|I_n|} \int f(t) h_n(t) dt \right) h_n$$

Deducimos que los funcionales duales asociados al sistema de Haar están dados por

$$h_n^* = \frac{1}{|I_n|} h_n, \quad n \in \mathbb{N}$$

y la serie de expansión de $f \in L_p$ en términos de la base de Haar es

$$f = \sum_{n=1}^{\infty} \left(\frac{1}{|I_n|} \int f(t) h_n(t) dt \right) h_n.$$

Observamos que si $p=2$, entonces $(h_n/\|h_n\|_2)_{n=1}^{\infty}$ es una base ortonormal para el espacio de Hilbert L_2 y por tanto incondicional. Este es un hecho importante, en realidad, la base de Haar es una base incondicional en L_p para $1 < p < \infty$. Esto fue probado por Paley en 1932. Mucho más recientemente, Burkholder [7] estableció la *mejor* constante. Vamos a presentar la demostración de Burkholder de 1988 [8]. Aunque aquí solamente haremos el caso real, la misma demostración funciona para escalares complejos con la misma constante; sin embargo, los cálculos necesarios para el caso complejo son un poco más duros de seguir. Para nuestros objetivos la constante no es tan importante, y simplemente

nos damos cuenta de que si la base de Haar es incondicional para escalares reales, rápidamente comprobamos que es incondicional para escalares complejos. La demostración de Burkholder sólo tiene un inconveniente: parece hecha por arte de magia.

Empezamos con algunos cálculos elementales.

2.3.5 Lema. *Supongamos $p > 2$. Entonces*

$$\frac{p^{p-2}}{(p-1)^{p-1}} < 1 \quad (2.9)$$

Demostración. Si cogemos $t = p - 1$ la desigualdad 2.9 es equivalente a

$$H(t) = -(t-1)\log(1+t) + t\log(t) > 0, \quad \forall t > 1.$$

En efecto, diferenciando H da

$$H'(t) = \frac{2}{t+1} - \log\left(\frac{1+t}{t}\right) \geq \frac{2}{t+1} - \frac{1}{t} = \frac{t-1}{t(t+1)} > 0, \quad \forall t > 1.$$

Por lo tanto $H(t) > H(1) = 0$ para todo $t > 1$. \square

En el siguiente lema introduciremos una función misteriosa que nos permitirá probar el teorema de Burkholder. Esta función parece ser sacada de la manga, aunque hay buenas razones detrás de su selección. El uso de tales funciones para probar desigualdades finas ha sido desarrollado extensamente por Nazarov, Treil y Volberg que las llamaron las *funciones de Belmberg*.

2.3.6 Lema. *Supongamos $p > 2$ y definimos una función φ en el primer cuadrante de \mathbb{R}^2 por*

$$\varphi(x, y) = (x+y)^{p-1}((p-1)x-y), \quad x, y \geq 0.$$

(I) *La siguiente desigualdad es válida para todo (x, y) con $x \geq 0$ e $y \geq 0$:*

$$\frac{(p-1)^{p-1}}{p^{p-2}} \varphi(x, y) \leq (p-1)^p x^p - y^p.$$

(II) *Para cualesquiera números reales x, y, a y para $\varepsilon = \pm 1$,*

$$\varphi(|x+a|, |y+\varepsilon a|) + \varphi(|x-a|, |y-\varepsilon a|) \geq 2\varphi(|x|, |y|). \quad (2.10)$$

Demostración. (I) Por homogeneidad podemos suponer que $x + y = 1$. Entonces es suficiente mostrar que la función

$$G(x) = \frac{p^{p-2}}{(p-1)^{p-1}} ((p-1)^p x^p - (1-x)^p) - px + 1, \quad 0 \leq x \leq 1,$$

es no negativa. Las primeras dos derivadas de G son

$$\begin{aligned} G'(x) &= \frac{p^{p-1}}{(p-1)^{p-1}}((p-1)^p x^{p-1} + (1-x)^{p-1}) - p, \\ G''(x) &= \frac{p^{p-1}}{(p-1)^{p-2}}((p-1)^p x^{p-2} + (1-x)^{p-2}). \end{aligned}$$

Como $p > 2$, G'' es creciente. Aún más,

$$G''(0) = -\frac{p^{p-1}}{(p-1)^{p-2}} < 0 < p((p-1)^2 - 1) = G''\left(\frac{1}{p}\right).$$

Por tanto, existe $0 < a < \frac{1}{p}$ tal que G es cóncava en $[0, a]$ y convexa en $[a, 1]$. Nos damos cuenta de que

$$G\left(\frac{1}{p}\right) = G'\left(\frac{1}{p}\right) = 0.$$

Consecuentemente, por la convexidad de G en $[a, 1]$, obtenemos que $G(x) > 0$ para todo $x \in [a, 1] \setminus \{\frac{1}{p}\}$. Uniendo esto con $G(a) > 0$ y aplicando 2.9 tenemos:

$$G(0) = 1 - \frac{p^{p-2}}{(p-1)^{p-1}} > 0.$$

Por lo tanto, por la concavidad de G en $[0, a]$, obtenemos $G(x) > 0$ para todo $x \in [0, a]$.

(II) El caso $\varepsilon = -1$ se puede deducir del caso $\varepsilon = 1$ reemplazando y por $-y$. Para probar la desigualdad con $\varepsilon = 1$ consideramos la familia de funciones

$$F_s : \mathbb{R} \rightarrow \mathbb{R}, t \mapsto \varphi(|t+s|, |t-s|),$$

definido para cada $s \in \mathbb{R}$. Entonces, para obtener 2.10 basta ver que F_s es una función convexa para cada s real. En efecto, si este fuera el caso, tendríamos:

$$F_s(t+a) + F_s(t-a) \geq 2F_s(t), \quad \forall t, a \in \mathbb{R},$$

y escogiendo s y t tal que $t+s = x$ y $t-s = y$, tendríamos la desigualdad deseada.

Para mostrar la convexidad de las funciones F_s , notamos que por un argumento de continuidad, es suficiente tratar con $s \neq 0$. Ahora, si $s \neq 0$, tenemos $F_s(t) = |s|^p F_1(\frac{t}{s})$. Por eso solo necesitamos probar que F_1 es una función convexa. Claramente, F_1 es continua en \mathbb{R} y diferenciable en $\mathbb{R} \setminus \{\pm 1\}$. Un cálculo sencillo nos da:

$$F'_1 = \begin{cases} p2^{p-1}(-t)^{p-2}((p-2)t - (p-1)) & \text{si } t < -1, \\ -p2^{p-1} & \text{si } -1 < t < 1, \\ p2^{p-1}t^{p-2}((p-2)t - (p-1)) & \text{si } 1 < t. \end{cases}$$

De aquí es claro que F'_1 es creciente en $(-\infty, -1)$ y constante en $(-1, 1)$. Diferenciando nuevamente en puntos con $t > 1$, obtenemos

$$F''_1(t) = p(p-1)(p-2)2^{p-1}(t-1)t^{p-2} > 0.$$

Por lo que F'_1 es también creciente en $(1, \infty)$. En los puntos frontera ± 1 , donde F_1 puede no ser suave,

$$F'_1((-1)^-) < F'_1((-1)^+) = F'_1(1^-) = F'_1(1^+).$$

Consecuentemente, F_1 es convexa en toda la recta real. \square

2.3.7 Teorema. Sean $0 < p, q < \infty$ de modo que $\frac{1}{p} + \frac{1}{q} = 1$ y sea $p^* = \max(p, q)$. La base de Haar $(h_n)_{n=1}^\infty$ en L_p es incondicional con constante incondicional como mucho $p^* - 1$. Esto es,

$$\left\| \sum_{n=1}^N \varepsilon_n a_n h_n \right\|_p \leq (p^* - 1) \left\| \sum_{n=1}^N a_n h_n \right\|_p,$$

siempre que $N \in \mathbb{N}$, para cualesquiera escalares reales a_1, \dots, a_N y cualesquiera elecciones de signos $\varepsilon_1, \dots, \varepsilon_N$.

Demostración. Supongamos primero $p > 2$, en cuyo caso $p^* = p$. Teniendo en cuenta el apartado (i) del Lema 2.3.6 solo necesitamos demostrar que

$$\int_0^1 \varphi \left(\left| \sum_{n=1}^N a_n h_n(u) \right|, \left| \sum_{n=1}^N \varepsilon_n a_n h_n(u) \right| \right) du \geq 0 \quad (2.11)$$

para todo $N \in \mathbb{N}$, todas las N -tuplas $(a_n)_{n=1}^N$ en \mathbb{R} , y todo $(\varepsilon_n)_{n=1}^N$ en $\{\pm 1\}$. Con ese fin, procedemos por inducción en N . Para $N = 1$ esto es trivial, ya que tomando $x = y = 0$ y $\varepsilon = 1$ en el apartado ii del Lema 2.3.6.

$$\varphi(|a|, |a|) \geq \varphi(0, 0) = 0, \quad \forall a \in \mathbb{R}.$$

Para establecer el paso a la inducción, supongamos $N \geq 2$ y asumimos que 2.11 es válido para $(N-1)$ tuplas. Dados $(a_n)_{n=1}^N$ en \mathbb{R} y $(\varepsilon_n)_{n=1}^N$ en $\{\pm 1\}$, consideramos

$$\begin{aligned} f_{N-1} &= \sum_{n=1}^{N-1} a_n h_n, \quad g_{N-1} = \sum_{n=1}^{N-1} \varepsilon_n a_n h_n \\ f_N &= \sum_{n=1}^N a_n h_n = f_{N-1} + a_N h_N, \quad g_N = \sum_{n=1}^N \varepsilon_n a_n h_n = g_{N-1} + \varepsilon_N a_N h_N \\ \tilde{f}_N &= \sum_{n=1}^N a_n h_n = f_{N-1} - a_N h_N, \quad \tilde{g}_N = \sum_{n=1}^N \varepsilon_n a_n h_n = g_{N-1} - \varepsilon_N a_N h_N. \end{aligned}$$

Dado que h_N tiene soporte en un intervalo en el cual f_{N-1} y g_{N-1} son constantes y toma valores opuestos en intervalos de la misma medida (o, usando terminología probabilística, h_N y h_{N-1} son variables aleatorias equidistribuidas, ambas independientes con la pareja de variables aleatorias (f_{N-1}, g_{N-1}) , tenemos que

$$J := \int_0^1 \varphi(|f_N(u)|, |g_N(u)|) du = \int_0^1 \varphi(|\tilde{f}_N(u)|, |\tilde{g}_N(u)|) du.$$

Por tanto, por el apartado II del Lema 2.3.6,

$$\begin{aligned} J &= \int_0^1 \frac{1}{2} \left(\varphi(|f_N(u)|, |g_N(u)|) + \varphi(|\tilde{f}_N(u)|, |\tilde{g}_N(u)|) \right) du \\ &\geq \int_0^1 \varphi(|f_{N-1}(u)|, |g_{N-1}(u)|) du \geq 0. \end{aligned}$$

El caso $p = 2$ es trivial, ya que el sistema de Haar es una base ortonormal de L_2 ; por lo tanto, su constante básica incondicional es 1. El caso $1 < p < 2$ se deduce ahora por dualidad. Con $f_N = \sum_{n=1}^{N-1} a_n h_n$ y $g_N = \sum_{n=1}^{N-1} \varepsilon_n a_n h_n$ como antes, elegimos $g'_N \in L_q(\mathcal{B}_N)$ tal que $\|g'_N\|_q = 1$ y

$$\int_0^1 g_N(u) g'_N(u) du = \|g_N\|_p.$$

Entonces $g'_N = \sum_{n=1}^N b_n h_n$ para alguna $(b_n)_{n=1}^N$. Sea $f'_N = \sum_{n=1}^N \varepsilon_n b_n h_n$. Es claro que

$$\|g_N\|_p = \int_0^1 f_N(u) f'_N(u) du \leq \|f_N\|_p \|f'_N\|_q \leq \|f_N\|_p (q-1) \|g'_N\|_q \leq (q-1) \|f_N\|_p. \quad \square$$

2.3.8 Nota. La constante $p^* - 1$ en el teorema de Burkholder es perfecta, aunque este hecho no lo vamos a probar.

2.3.9 Teorema. *Sea $1 \leq p < \infty$. Para cada conjunto finito de funciones $\{f_i\}_{i=1}^n$ en $L_p(\mu)$, se cumple que*

$$A_p \left\| \left(\sum_{i=1}^n |f_i|^2 \right)^{1/2} \right\|_p \leq \left(\mathbb{E} \left\| \sum_{i=1}^n \varepsilon_i f_i \right\|_p^p \right)^{1/p} \leq B_p \left\| \left(\sum_{i=1}^n |f_i|^2 \right)^{1/2} \right\|_p,$$

donde A_p , B_p son las constantes en las desigualdades de Khintchine (en particular, $A_p = 1$ para $2 \leq p < \infty$ y $B_p = 1$ para $1 \leq p \leq 2$).

Demostración. Para cada $\omega \in \Omega$, de las desigualdades de Khintchine,

$$A_p \left(\sum_{i=1}^n |f_i(\omega)|^2 \right)^{1/2} \leq \left(\mathbb{E} \left| \sum_{i=1}^n \varepsilon_i f_i(\omega) \right|^p \right)^{1/p},$$

donde $A_p = 1$ para $2 \leq p < \infty$. Ahora, usando el teorema de Fubini, obtenemos

$$A_p \left\| \left(\sum_{i=1}^n |f_i|^2 \right)^{1/2} \right\|_p^p \leq \int_{\Omega} \left\| \sum_{n=1}^n \varepsilon_i f_i(\omega) \right\|^p d\mu = \mathbb{E} \left(\int_{\Omega} \left| \sum_{n=1}^n \varepsilon_i f_i(\omega) \right|^p d\mu \right) = \mathbb{E} \left\| \sum_{i=1}^n \varepsilon_i f_i \right\|_p^p.$$

La otra desigualdad se obtiene de manera similar. \square

2.4. Bases de tipo greedy

Supongamos que X es un espacio de Banach y que $\mathcal{B} = (e_n)_{n=1}^\infty$ es una base de X . Una *aproximación de m-términos* con respecto a \mathcal{B} es una aplicación $T_m : X \rightarrow X$ tal que para cada $x \in X$, $T_m(x)$ es una combinación lineal de como mucho m elementos de \mathcal{B} . Un *algoritmo de aproximación* es una sucesión $(T_m)_{m=1}^\infty$ de tales aplicaciones.

El algoritmo de aproximación más natural es el *algoritmo lineal* $(S_m)_{m=1}^\infty$ dado por las proyecciones de las sumas parciales $S_m(x) = \sum_{j=1}^m e_j^*(x)e_j$. Para cada m , S_m proporciona la aproximación de m -términos más cercana para cada $x \in X$ del subespacio lineal $[e_1, \dots, e_m]$. Esto es, si para cada $x \in X$ definimos el *error de la mejor aproximación lineal de m-términos* como

$$E_m(x) = \inf \left\{ \left\| x - \sum_{j=1}^m \alpha_j e_j \right\| : (\alpha_j)_{j=1}^m \text{ escalares} \right\},$$

tenemos

$$\|x - S_m(x)\| \leq C E_m(x), \quad m = 1, 2, \dots,$$

para alguna constante $C \geq 1$ independiente de $x \in X$ y $m \in \mathbb{N}$. Así la teoría de aproximación lineal con respecto a bases resulta ser simple y conveniente si uno está contento con este nivel de precisión.

Motivados por el problema de encontrar algoritmos más eficientes (i.e., que mejoran el error de aproximación), los investigadores pusieron en juego la no linealidad permitiendo a los elementos usados en la aproximación depender del vector x al aproximarlos en vez de cogiéndolos de un espacio lineal fijado. Los correspondientes algoritmos de aproximación construidos de este modo definen aplicaciones que pueden no ser lineales o continuas. Konyagin y Telmyakov introdujeron en [9] el *algoritmo greedy* $(G_m)_{m=1}^\infty$, donde $G_m(x)$ se obtiene tomando los m primeros términos en orden decreciente de magnitud de la serie de expansión de x con respecto a \mathcal{B} ; cuando dos términos tienen igual tamaño, los tomamos en el orden de la base.

En términos generales, nuestro objetivo es investigar cómo de bien $G_m(x)$ aproxima $x \in X$ para varias bases \mathcal{B} . El mínimo requisito para hacer de este un método razonable

de aproximación es que $G_m(x)$ converja a x para cada $x \in X$. Nosotros vamos a exigir más a $G_m(x)$, queremos que nos proporcione la mejor aproximación de x por una suma de m elementos de la base, lo cual nos lleva a la noción de *base greedy*. La definición formal de base greedy fue dada por Konyagin y Temlyakov en [9].

2.4.1. Marco general

Sea X un espacio de Banach. A lo largo de este capítulo $\mathcal{B} = (e_n)_{n=1}^\infty$ será una base (de Schauder) seminormalizada en X (i.e., $1/c \leq \|e_n\| \leq c$ para cada n , para algún c) con funcionales biortogonales $(e_n^*)_{n=1}^\infty$.

Para cada $m = 1, 2, \dots$, denotamos por $\Sigma_m[\mathcal{B}, X]$ a la colección de todos los $x \in X$ que pueden ser expresados por una combinación lineal de m elementos de \mathcal{B} ,

$$\Sigma_m[\mathcal{B}, X] = \left\{ \sum_{n \in A} a_n e_n : A \subset \mathbb{N}, |A| = m, a_n \in \mathbb{R} \right\}.$$

Cuando la base \mathcal{B} , el espacio X , o ambos están claros por el contexto los suprimiremos de la notación anterior y usaremos solamente el símbolo Σ_m . Claramente $\Sigma_m \subset \Sigma_k$ siempre que $m \leq k$. Notemos que el espacio Σ_m no es lineal: la suma de dos elementos de Σ_m generalmente no está en Σ_m , está en Σ_{2m} .

La pregunta fundamental aquí es cómo podemos construir para cada $x \in X$ y cada $m = 1, 2, \dots$ un elemento $y_m \in \Sigma_m$, tal que el error de aproximación de x por y_m , dado por la cantidad $\|x - y_m\|$, es pequeño.

La respuesta a esta pregunta en algunos casos particulares es simple. Por ejemplo, si $X = H$ es un espacio de Hilbert y \mathcal{B} es una base ortonormal, cada elemento $x \in H$ tiene una expansión de la forma

$$x = \sum_{n=1}^{\infty} \langle x, e_n \rangle e_n,$$

y por la identidad de Parseval,

$$\|x\|^2 = \sum_{n=1}^{\infty} |\langle x, e_n \rangle|^2.$$

Uno se da rápidamente cuenta de que la mejor aproximación de x en Σ_m existe, y se obtiene como sigue. Ordenamos los coeficientes de Fourier $(\langle x, e_n \rangle)_{n=1}^\infty$ de x de acuerdo al valor absoluto de su tamaño, y escogemos $A_m(x)$ como el conjunto de índices n con cardinalidad m para los cuales $|\langle x, e_n \rangle|$ es el más grande. Entonces,

$$y_m = \sum_{n \in A_m(x)} \langle x, e_n \rangle e_n$$

es la mejor aproximación de x de Σ_m , y

$$\|x - y_m\|^2 = \sum_{n \notin A_m(x)} |\langle x, e_n \rangle|^2.$$

Este es un ejemplo de lo que es conocido como *algoritmo greedy*. Existen dos caminos esencialmente equivalentes de formalizar esta idea y simultáneamente generalizarla a bases seminormalizadas en un espacio de Banach arbitrario X .

La primera es considerar para cada $\varepsilon > 0$ el *operador umbral*,

$$T_\varepsilon[\mathcal{B}, X](x) := T_\varepsilon(x) = \sum_{\{n : |e_n^*(x)| > \varepsilon\}} e_n^*(x) e_n.$$

Como $\lim_{n \rightarrow \infty} e_n^* = 0$, estamos sumando solo un número finito de términos, y por tanto las aplicaciones $(T_\varepsilon)_{\varepsilon > 0}$ están bien definidas.

En el segundo enfoque fijamos $m \in \mathbb{N}$, y para $x \in X$ definimos la *suma greedy* de x de orden m como

$$G_m[\mathcal{B}, X](x) := G_m(x) = \sum_{n \in A_m(x)} e_n^*(x) e_n,$$

donde $A_m(x)$ es un conjunto de m -elementos de índices tales que

$$\min\{|e_n^*(x)| : n \in A_m(x)\} \geq \max\{|e_n^*(x)| : n \notin A_m(x)\}. \quad (2.12)$$

Por supuesto, los conjuntos $A_m(x)$ pueden no estar determinados únicamente por las condiciones previas; por lo tanto, un elemento $x \in X$ dado puede tener más de una suma greedy de cualquier orden. Sin embargo, nos damos cuenta de que si en 2.12 se da la desigualdad estricta entonces $\sum_{n \in A_m(x)} e_n^*(x) e_n$ es la única suma greedy de x de orden m .

Sumas estrictamente greedy y operadores umbrales están estrechamente relacionados. A cada $x \in X$ le corresponde un subconjunto de enteros

$$\mathbb{N}_x = \{m \in \mathbb{N} : \text{existe una suma estrictamente greedy de } x \text{ de orden } m\}.$$

Entonces para cada $\varepsilon > 0$ y $m \in \mathbb{N}_x$, tenemos

$$T_\varepsilon(x) = G_{m_\varepsilon}(x) \quad \text{y} \quad G_m(x) = T_{\varepsilon_m}(x), \quad (2.13)$$

donde

$$m_\varepsilon = |\{n \in \mathbb{N} : |e_n^*(x)| > \varepsilon\}| \quad \text{y} \quad \max_{n \notin A_m(x)} |e_n^*(x)| \leq \varepsilon < \min_{n \in A_m(x)} |e_n^*(x)|.$$

La manera más natural de construir una suma greedy de un vector x es comenzar con una aplicación inyectiva $\pi : \mathbb{N} \rightarrow \mathbb{N}$ tal que $(|e_{\pi(n)}^*(x)|)_{n=1}^\infty$ no es creciente y entonces considerar las sumas parciales

$$G_m(x) = \sum_{n=1}^m e_{\pi(n)}^*(x) e_{\pi(n)}$$

de las series formales $\sum_{n=1}^\infty e_{\pi(n)}^*(x) e_{\pi(n)}$. Toda sucesión $(G_m(x))_{m=1}^\infty$ obtenida así la llamaremos *aproximación greedy* de x , y a π lo llamaremos *ordenamiento greedy* de x . Si el soporte de x es finito, entonces también lo es \mathbb{N}_x ; por lo tanto se puede enumerar en la forma $\mathbb{N}_x = (m_j)_{j=1}^\infty$ con los índices m_j crecientes. La sucesión $(G_{m_j}(x))_{j=1}^\infty$ es una sub-sucesión de cada aproximación greedy x , y la llamaremos la *aproximación estrictamente greedy* de x .

La aproximación greedy de un vector x no tiene por qué ser única. Sin embargo, podemos usar el orden natural existente en \mathbb{N} para construir para cada x un orden greedy únicamente determinado como sigue. Definimos $\rho : \mathbb{N} \rightarrow \mathbb{N}$ tal que $\{n : e_n^*(x) \neq 0\} \subset \rho(\mathbb{N})$ y tal que si $j < k$, entonces o bien $|e_{\rho(j)}^*(x)| > |e_{\rho(k)}^*(x)|$ o $|e_{\rho(j)}^*(x)| = |e_{\rho(k)}^*(x)|$ y $\rho(j) < \rho(k)$. Con esta convención, la m -ésima suma greedy de x , dada por

$$\mathcal{G}_m[\mathcal{B}, X](x) := \mathcal{G}_m(x) = \sum_{n=1}^m e_{\rho(n)}^*(x) e_{\rho(n)},$$

está determinada de manera única, y $(\mathcal{G}_m(x))_{m=1}^\infty$ forma una aproximación greedy de x . Las sucesiones de aplicaciones $(\mathcal{G}_m)_{m=1}^\infty$ son el *algoritmo greedy* asociado a \mathcal{B} en X . En algunas ocasiones nuestro razonamiento requerirá considerar las sumas greedy de orden nulo, así que acordamos poner $\mathcal{G}_0(x) = 0$ para todo x .

Nos damos cuenta de que $(\mathcal{G}_m)_{m=1}^\infty$ no son ni lineales ni continuos. Sin embargo, son homogéneos, es decir, $\mathcal{G}_m(\lambda x) = \lambda \mathcal{G}_m(x)$ para $\lambda \in \mathbb{R}$, gracias a lo que podemos definir su norma,

$$\|\mathcal{G}_m\| = \sup_{\|x\| \leq 1} \|\mathcal{G}_m(x)\|.$$

Notamos que si $T : X \rightarrow Y$ es un isomorfismo entre espacios de Banach, entonces

$$T(\mathcal{G}_m[\mathcal{B}, X](x)) = \mathcal{G}_m[T(\mathcal{B}), Y](T(x));$$

por lo que todos los conceptos relacionados con ser greedy que vamos a introducir a continuación serán invariantes bajo isomorfismos.

2.4.2. Bases democráticas

Vamos a introducir a continuación una propiedad que disfrutan algunas bases que es fundamental para nuestras consideraciones [9].

2.4.1 Definición. Una base $\mathcal{B} = (e_n)_{n=1}^{\infty}$ de un espacio de Banach se dice *democrática* si bloques del mismo tamaño de \mathcal{B} tienen normas uniformemente comparables, i.e., existe una constante $C \geq 1$ tal que para cada dos subconjuntos finitos A, B de \mathbb{N} con $|A| = |B|$,

$$\left\| \sum_{n \in A} e_n \right\| \leq C \left\| \sum_{n \in B} e_n \right\|.$$

La constante más pequeña verificando lo anterior se llama la *constante democrática* de \mathcal{B} y la denotaremos por C_d .

La falta de democracia de una base \mathcal{B} exhibe algún tipo de asimetría. Para medir cuánto se desvía una base de ser democrática, consideramos su *función democrática superior*, también conocida como la *función fundamental* de \mathcal{B} ,

$$\varphi_u[\mathcal{B}, X](m) := \varphi_u(m) = \sup_{|A|=m} \left\| \sum_{n \in A} e_n \right\|, \quad m = 1, 2, \dots,$$

y su *función democrática inferior*,

$$\varphi_l[\mathcal{B}, X](m) := \varphi_l(m) = \inf_{|A|=m} \left\| \sum_{n \in A} e_n \right\|, \quad m = 1, 2, \dots$$

Una base \mathcal{B} es democrática si y solo si las sucesiones $(\varphi_u[\mathcal{B}, X](m))_{m=1}^{\infty}$ y $(\varphi_l[\mathcal{B}, X](m))_{m=1}^{\infty}$ son uniformemente comparables término a término, en cuyo caso

$$C_d = \sup_{m \in \mathbb{N}} \frac{\varphi_u(m)}{\varphi_l(m)} < \infty.$$

Por supuesto, las funciones democráticas $\varphi_l[\mathcal{B}, X]$ y $\varphi_u[\mathcal{B}, X]$ pueden variar si consideramos diferentes bases \mathcal{B} dentro del mismo espacio de Banach X .

Los siguientes resultados contienen propiedades elementales de funciones democráticas de bases que nos serán útiles más tarde.

2.4.2 Lema. Sea \mathcal{B} una base democrática en un espacio de Banach X y sea $\tilde{\mathcal{B}}$ una base de un espacio de Banach finito-dimensional Y . La suma directa de \mathcal{B} y $\tilde{\mathcal{B}}$ es una base democrática en $X \oplus Y$.

El último resultado de este apartado relaciona la incondicionalidad con las funciones de democracia.

2.4.3 Lema. Sea $(e_n)_{n=1}^{\infty}$ una base incondicional para un espacio de Banach X . Sean A y B conjuntos finitos de naturales. Supongamos que $x \in X$ es tal que

$$\max\{|e_n^*(x)| : n \in A\} \leq \min\{|e_n^*(x)| : n \in B\}.$$

Entonces

$$\|\mathbf{P}_A(x)\| \leq K_u^2 \frac{\varphi_u(|A|)}{\varphi_l(|B|)} \|\mathbf{P}_B(x)\|.$$

Demostración. Sean $M = \max\{|e_n^*(x)| : n \in A\}$ y $m = \min\{|e_n^*(x)| : n \in B\}$. Entonces

$$\|\mathbf{P}_A(x)\| = \left\| \sum_{n \in A} e_n^*(x) e_n \right\| \leq K_u \left\| \sum_{n \in A} M e_n \right\| = K_u M \left\| \sum_{n \in A} e_n \right\| \leq K_u M \varphi_u(|A|)$$

donde hemos usado la incondicionalidad y la definición de la función democrática superior. Ahora continuamos usando la hipótesis $M \leq m$ y la definición de la función democrática inferior:

$$K_u M \varphi_u(|A|) \leq K_u m \frac{\varphi_u(|A|)}{\varphi_l(|B|)} \varphi_l(|B|) \leq K_u \frac{\varphi_u(|A|)}{\varphi_l(|B|)} \left\| \sum_{n \in B} m e_n \right\| \leq K_u^2 \frac{\varphi_u(|A|)}{\varphi_l(|B|)} \|\mathbf{P}_B(x)\|,$$

lo que concluye la demostración. \square

2.4.3. Bases greedy

En la teoría de la aproximación es conveniente saber desde un punto teórico si el algoritmo greedy es eficiente, en el sentido de que el error que cometemos para cada m al aproximar cualquier x en X por $\mathcal{G}_m(x)$ es uniformemente comparable con el error teórico más pequeño en la aproximación de m -términos de x con respecto a la base B , dado por

$$\sigma_m[\mathcal{B}, X](x) := \sigma_m(x) = \inf_{y \in \Sigma_m} \|x - y\|,$$

y para una clase de funciones F el error teórico es

$$\sigma_m[\mathcal{B}, X](F) := \sup_{f \in F} \sigma_m[\mathcal{B}, X](f).$$

Para formalizar esta idea, Konyagin y Temlyakov introdujeron el concepto de base greedy en [9].

2.4.4 Definición. Una base \mathcal{B} de un espacio de Banach X es *greedy* si existe una constante absoluta $C \geq 1$ tal que

$$\|x - \mathcal{G}_m(x)\| \leq C \sigma_m(x), \quad \forall m \in \mathbb{N}, \quad \forall x \in X. \tag{2.14}$$

A la C más pequeña la llamaremos la constante *greedy* de \mathcal{B} y la denotaremos por C_g .

Observamos que $0 \in \Sigma_m$ para todo m , por lo que tenemos

$$\sigma_m(x) \leq \|x\|, \quad \forall x \in X.$$

También como $\cup_{n=1}^{\infty} \Sigma_n$ es denso en X , entonces

$$\lim_{m \rightarrow \infty} \sigma_m(x) = 0.$$

2.4.5 Nota. Si hemos estado atentos, la definición de base greedy es equivalente al cumplimiento de la condición

$$\|x - G_m(x)\| \leq C \sigma_m(x), \quad \forall m \in \mathbb{N}, \quad \forall x \in X$$

para alguna constante C , para cualesquiera sumas greedy $G_m(x)$.

2.4.6 Ejemplo. Si \mathcal{B} es la base de vectores unitarios en ℓ_p ($1 \leq p < \infty$) o c_0 , entonces para cada x en el espacio y cualquier entero m tenemos $\sigma_m(x) = \|x - G_m(x)\|$ para cada suma greedy de m -términos $G_m(x)$, la cual es lo mejor que podemos esperar. En otras palabras, \mathcal{B} es una base greedy y $C_g = 1$. Esta propiedad puede ser extendida a las bases simétricas.

El principal resultado de esta sección es una caracterización intrínseca muy satisfactoria de Konyagin y Temlyakov de las bases greedy.

2.4.7 Teorema. Una base $\mathcal{B} = (e_n)_{n=1}^{\infty}$ de un espacio de Banach X es greedy si y solo si es incondicional y democrática.

Demostración. Asumimos que \mathcal{B} es una base greedy con constante greedy C_g . Fijamos $x \in X$ y sea $A \subset \mathbb{N}$ con cardinalidad m . Consideramos el vector

$$y = P_{A^c}(x) + \alpha \sum_{n \in A} e_n = x + \sum_{n \in A} (\alpha - e_n^*(x)) e_n,$$

donde $\alpha > \sup_{n \notin A} |e_n^*(x)|$. Claramente, $\sigma_m(y) \leq \|x\|$ y $\mathcal{G}_m(y) = \alpha \sum_{n \in A} e_n$. Por lo tanto, la suposición de ser greedy nos dice

$$\|P_{A^c}(x)\| = \|y - \mathcal{G}_m(y)\| \leq C_g \sigma_m(y) \leq C_g \|x\|,$$

lo cual implica que \mathcal{B} es incondicional con constante de supresión-incondicional acotada por C_g .

Para mostrar que \mathcal{B} es democrática, cogemos cualesquiera dos conjuntos de enteros A, B del mismo cardinal. Consideramos $x = \sum_{n \in A \cup B} e_n$. Como $\sum_{n \in B \setminus A} e_n$ es la suma greedy de x de orden $m = |A \setminus B| = |B \setminus A|$, y $\sum_{n \in A \setminus B} e_n \in \Sigma_m$, tenemos

$$\left\| \sum_{n \in A} e_n \right\| = \left\| x - \sum_{n \in B \setminus A} e_n \right\| \leq C_g \left\| x - \sum_{n \in A \setminus B} e_n \right\| = C_g \left\| \sum_{n \in B} e_n \right\|,$$

por lo que \mathcal{B} es democrática con constante democrática acotada por C_g .

Para demostrar la otra implicación, asumimos que \mathcal{B} es incondicional con constante K_u y democrática con constante democrática C_d . Fijamos $x \in X$ y $m \in \mathbb{N}$. Cogemos $y = \sum_{n \in B} a_n e_n \in \Sigma_m$. Consideramos una suma greedy $G_m(x) = P_A(x)$. Notemos que

$$\|x - G_m(x)\| = \|P_{(A \cup B)^c}(x) + P_{B \setminus A}(x)\| \leq \|P_{(A \cup B)^c}(x)\| + \|P_{B \setminus A}(x)\|.$$

Por la definición de suma greedy,

$$\max_{n \in B \setminus A} |e_n^*(x)| \leq \min_{n \in A \setminus B} |e_n^*(x)|.$$

Sea $r = |A \setminus B| = |B \setminus A|$. Usando el Lema 2.4.3, tenemos

$$\|P_{B \setminus A}(x)\| \leq K_u^2 \frac{\varphi_u(r)}{\varphi_l(r)} \|P_{A \setminus B}(x)\| \leq K_u^2 C_d \|P_{A \setminus B}(x)\|.$$

La incondicionalidad de \mathcal{B} implica

$$\|P_{(B \cup A)^c}(x)\| = \|P_{(B \cup A)^c}(x - y)\| \leq K_{su} \|x - y\|.$$

Análogamente,

$$\|P_{A \setminus B}(x)\| = \|P_{A \setminus B}(x - y)\| \leq K_{su} \|x - y\|.$$

Combinándolos, obtenemos

$$\|x - G_m(x)\| \leq (K_{su} + K_{su} K_u^2 C_d) \|x - y\|,$$

y, con el Lema 2.4.3 en mente, la prueba finaliza tomando el ínfimo en y . \square

2.4.8 Corolario. Toda base subsimétrica en un espacio de Banach es greedy.

Demostración. Supongamos que $\mathcal{B} = (e_n)_{n=1}^\infty$ es subsimétrica y no greedy. Como \mathcal{B} es incondicional, debe fallar que sea democrática. Recursivamente construimos sucesiones de subconjuntos mutuamente disjuntos de enteros $(A_k)_{k=1}^\infty$ y $(B_k)_{k=1}^\infty$ tal que $|A_k| = |B_k|$, $\max(A_{k-1} \cup B_{k-1}) < \min(A_k \cup B_k)$, y $\left\| \sum_{n \in A_k} e_n \right\| > k \left\| \sum_{n \in B_k} e_n \right\|$. Supongamos construidos $(A_j)_{j=1}^{k-1}$ y $(B_j)_{j=1}^{k-1}$. Establecemos $N = 1 + \max(A_{k-1} \cup B_{k-1})$. Usando el Lema 2.4.2, inferimos que la sucesión básica $(e_n)_{n=N}^\infty$ no es democrática. Por tanto, existen A_k , B_k subconjuntos de $\{n \in \mathbb{N} : n \geq N\}$ tal que $|A_k| = |B_k|$ y $\left\| \sum_{n \in A_k} e_n \right\| > k \left\| \sum_{n \in B_k} e_n \right\|$.

Ahora consideraremos sucesiones estrictamente crecientes de enteros $(n_j^a)_{j=1}^\infty$ y $(n_j^b)_{j=1}^\infty$ tales que $\{n_j^a : j \in \mathbb{N}\} = \cup_{k=1}^\infty A_k$ y $\{n_j^b : j \in \mathbb{N}\} = \cup_{k=1}^\infty B_k$. Como las sucesiones básicas $(e_{n_j^a})_{j=1}^\infty$ y $(e_{n_j^b})_{j=1}^\infty$ no son equivalentes, alcanzamos una contradicción con que \mathcal{B} sea subsimétrica. \square

El siguiente lema será una herramienta importante para la demostración de la Proposición 2.4.10.

2.4.9 Lema. *Sea $1 < r < \infty$ y $0 < p < \infty$. Existen constantes positivas $c_{r,p}$ y $C_{r,p}$ tales que para cada conjunto finito de enteros A ,*

$$c_{r,p} \left(\sum_{k \in A} r^{pk} \right)^{1/p} \leq \left(\sum_{k \in A} r^{2k} \right)^{1/2} \leq C_{r,p} \left(\sum_{k \in A} r^{pk} \right)^{1/p}.$$

Después del arduo trabajo que nos ha tomado llegar hasta aquí, estamos en condiciones de enunciar el resultado más importante del trabajo. Fue obtenido por Temlyakov en [10] y nos permitirá usar el sistema de Haar como herramienta fundamental para la aproximación en la parte informática.

2.4.10 Proposición. *El sistema de Haar normalizado $\mathcal{H}_p = (h_n^p)_{n=1}^\infty$ es una base greedy en $L_p[0, 1]$ para $1 < p < \infty$.*

Demostración. Como \mathcal{H}_p es incondicional (Teorema 2.3.7) por el Teorema 2.4.7 sólo necesitamos probar que \mathcal{H}_p es democrática. Para probar esto, por el Lema 2.4.2 será suficiente estimar $\|\sum_{n \in A} h_n^p\|$ por subconjuntos finitos de enteros A tales que $1 \notin A$.

Tenemos en cuenta que las funciones $(h_n^p)_{n=2}^\infty$ solo alcanzan valores 0 y $\pm 2^{k/p}$, y que para cada $t \in [0, 1]$ sólo una de esas funciones alcanza $2^{k/p}$ en valor absoluto. Usando la propiedad de las series geométricas enunciada en el Lema 2.4.9 con $r = 2^{1/p}$ obtenemos constantes c_p y C_p de manera que para cada subconjunto finito $A \subset \mathbb{N} \setminus \{1\}$ y para cada $t \in [0, 1]$, se cumple que

$$c_p \left(\sum_{n \in A} |h_n^p(t)|^p \right)^{1/p} \leq \left(\sum_{n \in A} |h_n^p(t)|^2 \right)^{1/2} \leq C_p \left(\sum_{n \in A} |h_n^p(t)|^p \right)^{1/p}.$$

Tomando L_p -normas en las desigualdades anteriores y usando que $\|h_n^p\|_p = 1$ obtenemos

$$c_p |A|^{1/p} \leq \left\| \left(\sum_{n \in A} |h_n^p(t)|^2 \right)^{1/2} \right\|_p \leq C_p |A|^{1/p}. \quad (2.15)$$

Por el Teorema 2.3.9 y la incondicionalidad de $\mathcal{H}_p = (h_n^p)_{n=1}^\infty$, para cada $1 < p < \infty$ existen constantes A'_p y B'_p tales que para cada $(a_n)_{n=1}^\infty \in c_{00}$, se tiene

$$A'_p \left\| \left(\sum_{n=1}^\infty |a_n|^2 |h_n^p|^2 \right)^{1/2} \right\|_p \leq \left\| \sum_{n=1}^\infty a_n h_n^p \right\|_p \leq B'_p \left\| \left(\sum_{n=1}^\infty |a_n|^2 |h_n^p|^2 \right)^{1/2} \right\|_p. \quad (2.16)$$

Combinando esto con 2.15 obtenemos

$$A'_p c_p |A|^{1/p} \leq \left\| \sum_{n \in A} h_n^p \right\| \leq B'_p C_p |A|^{1/p},$$

para cada $A \subset \mathbb{N} \setminus \{1\}$ finito, lo cual implica la democracia de $(h_n^p)_{n=2}^\infty$. \square

Los ejemplos naturales de bases greedy en los espacios clásicos que hemos visto en este apartado pueden ser útiles para encontrar bases greedy cuando están combinados con métodos teóricos simples. Observemos el hecho de que ser greedy es una propiedad isomórfica, i.e., si $(e_n)_{n=1}^\infty$ es una base greedy en X y $T : X \rightarrow Y$ es un isomorfismo lineal, entonces $(T(e_n))_{n=1}^\infty$ es una base greedy en Y .

2.4.4. Algoritmos greedy teóricos

Como culmen a todo el trabajo realizado se van a estudiar algoritmos greedy teóricos desarrollados por Temlyakov en [11]. A partir de este momento se va intentar dar un enfoque más computacional.

Buscamos un algoritmo de construcción de m -términos que añade en cada paso un nuevo elemento del sistema dado \mathcal{D} . El sistema \mathcal{D} de X es la base aunque en este enfoque más informático lo denominamos *diccionario* y verifica que cada $g \in \mathcal{D}$ tiene norma uno y su cierre lineal coincide con X . No es obvio que con este nivel de generalidad (X y \mathcal{D} son arbitrarios) tal algoritmo exista.

Algoritmo Puro Greedy (AGP)

Comencemos esta discusión con un caso especial, un espacio de Hilbert con el producto escalar $\langle \cdot, \cdot \rangle$. Definimos en primer lugar el Algoritmo Puro Greedy (AGP) en espacios de Hilbert H . Describiremos este algoritmo para un diccionario general \mathcal{D} . Si $f \in H$, sea $g(f) \in \mathcal{D}$ un elemento de \mathcal{D} que maximiza $|\langle f, g \rangle|$. Asumiremos por simplicidad que tal maximizador existe; si no modificaciones adecuadas son necesarias (Algoritmo Greedy Débil (AGD)) en el algoritmo que sigue. Definimos

$$\begin{aligned} G[\mathcal{D}, H](f) &:= G(f) = \langle f, g(f) \rangle g(f), \\ R[\mathcal{D}, H](f) &:= R(f) = f - G(f). \end{aligned}$$

Podemos omitir en la notación el diccionario y el espacio cuando sean evidentes.

Algoritmo Greedy Puro (AGP). Definimos $R_0[\mathcal{D}, H](f) := f$ y $G_0[\mathcal{D}, H](f) := 0$. Entonces, para cada $m \geq 1$, definimos inductivamente

$$\begin{aligned} G_m[\mathcal{D}, H](f) &:= G_m(f) = G_{m-1}(f) + G(R_{m-1}(f)), \\ R_m[\mathcal{D}, H](f) &:= R_m(f) = f - G_m(f) = R(R_{m-1}(f)). \end{aligned}$$

El *algoritmo greedy* definido da un procedimiento para construir un aproximante que resulta ser bueno. El procedimiento para construir un aproximante greedy no es un algoritmo numérico listo para implementación computacional. Por tanto, sería más preciso llamarlo *algoritmo greedy teórico*. Llamamos un algoritmo *incremental* si en el paso m añadimos a lo mucho un elemento más, $\varphi_m \in \mathcal{D}$ y aproximamos por la combinación lineal $c_1 \varphi_1 + \cdots + c_m \varphi_m$.

Usamos el término *tipo greedy* para un algoritmo incremental con φ_m escogido para maximizar un funcional dado $F(f_{m-1}, g)$ sobre $g \in \mathcal{D}$ con f_{m-1} el residuo después de $(m-1)$ pasos del algoritmo. La forma de $F(\cdot, \cdot)$ determina el tipo de algoritmo greedy. Usaremos el término *greedy débil* para un algoritmo incremental con φ_m satisfaciendo una condición más débil que maximizar un funcional dado. Por ejemplo,

$$F(f_{m-1}^\tau, \varphi_m) \geq t_m \sup_{g \in \mathcal{D}} F(f_{m-1}, g), \quad 0 \leq t_m \leq 1.$$

A la sucesión $\tau := \{t_k\}_{k=1}^\infty$ la llamaremos la *sucesión de debilidad*.

También anotar que en otros campos de la investigación AGP recibe otros nombres, en procesamiento de señales recibe el nombre de *matching pursuit*.

Es claro que para una base ortonormal \mathcal{B} de un espacio de Hilbert H tenemos, para cada f ,

$$\|f - G_m[\mathcal{B}, H](f)\| = \sigma_m[\mathcal{B}, H](f).$$

Existe una clase no trivial de ejemplos de un diccionario redundante, teniendo la misma propiedad: AGPs realizan la mejor aproximación de m -términos para cada función individual. Describimos ese diccionario ahora. Sea Π un conjunto de funciones de $H = L_2([0, 1]^2)$ de la forma $u(x_1)v(x_2)$ con la norma unitaria de L_2 . Entonces para este diccionario y para cada $f \in H$,

$$\|f - G_m[\Pi, H](f)\| = \sigma_m[\Pi, H](f).$$

Algoritmo X-Greedy

Sabemos mucho menos sobre algoritmos greedy en el caso de un espacio general de Banach X . Vamos a discutir una generalización de AGP de un espacio de Hilbert H a un espacio de Banach X . Llamamos a esta versión el Algoritmo X -Greedy cuando queremos indicar el espacio de Banach. Para un X y \mathcal{D} dado definimos $G[\mathcal{D}, X](f) := \alpha(f)g(f)$ donde $\alpha(f) \in \mathbb{R}$ y $g(f) \in \mathcal{D}$ satisfacen (asumimos existencia) la relación

$$\min_{\alpha \in \mathbb{R}, g \in \mathcal{D}} \|f - \alpha(f)g(f)\|.$$

Algoritmo X -Greedy. Definimos $R_0[\mathcal{D}, X](f) := f$ y $G_0[\mathcal{D}, X](f) := 0$. Entonces, para cada $m \geq 1$, inductivamente definimos

$$\begin{aligned} G_m[\mathcal{D}, X](f) &:= G_m(f) = G_{m-1}(f) + G(R_{m-1}(f)), \\ R_m[\mathcal{D}, X](f) &:= R_m(f) = R_{m-1}(f) - G(R_{m-1}(f)). \end{aligned}$$

La extensión de AGP a un espacio de Banach se conoce como el Algoritmo Greedy Dual (AGD). Sea un diccionario \mathcal{D} en X dado, tomamos un elemento $f \in X$ y encontramos un funcional F_f tal que $\|F_f\|_{X^*} = 1$ y $F_f(f) = \|f\|_X$. Este funcional es un *funcional normante* de f y su existencia se sigue del Teorema de Hanh-Banach. Ahora modificamos el paso básico de un AGP a lo siguiente. Asumimos que existe $g_f \in \mathcal{D}$ tal que

$$|F_f(g_f)| = \max_{g \in D} |F_f(g)|.$$

Tomamos esta g_f y resolvemos un problema de optimización más: encontrar el número a tal que

$$\|f - ag_f\|_X = \min_b \|f - bg_f\|_X.$$

Por último,

$$G[\mathcal{D}, X](f) := ag_f, \quad R[\mathcal{D}, X](f) := f - ag_f.$$

Repitiendo este paso m veces tenemos $G_m(f)$ como un aproximante y $R_m(f)$ como un residuo.

Capítulo 3

Desarrollo informático del trabajo

El aspecto informático del trabajo está basado en teoría de aproximación, que es el nexo con el aspecto matemático. En este caso se estudia cómo una función puede ser aproximada por otras funciones más simples, incluyendo una caracterización cuantitativa del error.

Los modelos teóricos existentes no son implementables. Por eso el primer objetivo de este capítulo es diseñar un procedimiento implementable de dichos modelos. La Sección 3.1 está dedicada a este objetivo.

Después de este diseño base buscaremos ejemplos donde testear la aproximación con la finalidad de hacer de ésta un método de compresión de información. Los ejemplos serán señales e imágenes. A cada ejemplo se dedica una sección para estudiar sus peculiaridades y las optimizaciones de parámetros que permite.

Todo el proyecto está disponible en: <https://github.com/Carlossamu7/TFG>.

3.1. De los algoritmos teóricos a la implementación

La idea de reemplazar un objeto completo por uno más simple (aproximante) es usada con éxito en muchas áreas de la ciencia y en particular en las matemáticas computacionales. Las matemáticas computacionales son el conjunto de programas de ordenador, algoritmos, técnicas y teorías necesarios para resolver en un ordenador modelos matemáticos de problemas que surgen en la ciencia y la tecnología. Son aquellas matemáticas donde la informática juega un papel esencial a través de los algoritmos y métodos numéricos.

3.1.1. Marco

Los objetos que van a ser aproximados son funciones. Un ejemplo pueden ser las señales pero no es difícil trasladarlo a imágenes y vídeo con las consideraciones adecuadas. Las funciones que estamos interesados en aproximar son finito dimensionales. Sin embargo, queremos tener un control en la precisión de la aproximación. Es por ello que el desarrollo se ha llevado a cabo en espacios de dimensión infinita.

El espacio euclídeo \mathbb{R}^n se puede generalizar a dimensión infinita a través del concepto de espacio de Hilbert. Como ejemplos de espacio de Hilbert tenemos ℓ_2 y $L_2[0, 1]$.

Dependiendo de los problemas que se quieran abordar puede resultar útil cambiar la forma de medir distancias por lo que existe un concepto más abstracto que el de espacio de Hilbert y es el de espacio de Banach. La característica principal de un espacio de Banach es que posee una estructura de espacio vectorial y de espacio métrico que permite medir distancias entre los objetos. Estas dos estructuras están bien relacionadas. La otra característica primordial de los espacios de Banach es la completitud, que a efectos prácticos quiere decir que cualquier operación de paso al límite devuelve un objeto del espacio. Todo nuestro trabajo se va a desarrollar en los espacios de funciones integrables $L_p[0, 1]$.

Bajo este ambiente de trabajo se ha realizado un estudio con el propósito de que dada una función f en un espacio de Banach X podamos construir

$$\mathcal{G}_m(f) = \sum_{g \in \Lambda} c_g g, \quad \Lambda \subset \mathcal{D}, \quad \#\Lambda \leq m,$$

la aproximación greedy de m -términos de f . Las funciones más simples que sirven para aproximar f son de una colección prefijada, la base greedy. Aquí la notamos como \mathcal{D} y nos referiremos a ella como diccionario pues es posible realizar este procedimiento teórico sin necesidad de que el conjunto \mathcal{D} sea una base del espacio.

El estudio analítico llevado a cabo en el desarrollo matemático nos proporciona el resultado fundamental para la aproximación. Este resultado es la Proposición 2.4.10 que dice que el sistema de Haar normalizado es greedy en $L_p[0, 1]$ ($1 < p < \infty$).

Aunque hay más bases greedy aquí se va a usar la de Haar. Así $\mathcal{D} = \mathcal{H}_p = (h_n^p)_{n=1}^\infty$, y por tanto nuestra búsqueda se puede expresar ahora de la siguiente manera:

$$\mathcal{G}_m(f) = \sum_{k \in A} c_k h_k, \quad A \subset \mathbb{N}, \quad \#A \leq m,$$

Tal y como vimos en la Sección 2.4.4 la búsqueda de los coeficientes es un proceso costoso y poco implementable. En este trabajo se ha construido un mecanismo de obtención de los coeficientes. Además el mecanismo resulta entendible, fácilmente implementable y eficiente.

3.1.2. Transformada de Haar discreta

La idea para hacer una compresión basada en esta teoría es discretizar el sistema de Haar. Mediante esta discretización se va a construir un procedimiento para arrays unidimensionales a los que nos referiremos más cómodamente como listas.

Discretizar el sistema de Haar a una lista de tamaño 2^n es tan sencillo como dividir en 2^n trozos el intervalo $[0, 1]$, dominio de las funciones de Haar, y asignarle a cada elemento de la lista el valor de la función de Haar en su respectivo trozo.

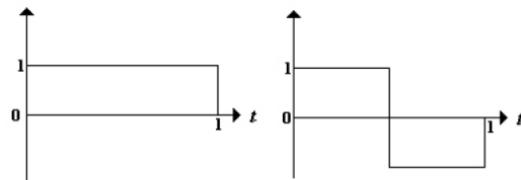


Imagen 3.1: Dos primeras funciones del sistema de Haar

Por ejemplo, para tamaño 4 las dos primeras funciones de Haar de la Imagen 3.1 quedarían discretizadas como (1 1 1 1) y (1 1 -1 -1).

Para entender el cálculo de la transformada de Haar sobre listas vamos a examinar el procedimiento para una lista concreta: (12 12 12 12 8 8 10 10). Hacemos las medias por parejas y obtenemos cuatro valores: $(12 + 12)/2 = 12$, $(12 + 12)/2 = 12$, $(8 + 8)/2 = 8$ y $(10 + 10)/2 = 10$. Con ellos nos resultaría imposible recuperar la lista original, por lo que vamos a calcular también las diferencias divididas por dos por parejas: $(12 - 12)/2 = 0$, $(12 - 12)/2 = 0$, $(8 - 8)/2 = 0$ y $(10 - 10)/2 = 0$. Así obtenemos cuatro medias y cuatro diferencias que presentamos en la siguiente lista: (12 12 8 10 0 0 0 0).

Ahora la segunda mitad de nuestra lista queda fija y aplicamos de manera recursiva este procedimiento a la primera mitad. Obtenemos (12 9 0 -1 0 0 0 0). En la última iteración tenemos fijados los últimos 6 valores y operamos sobre los dos primeros obteniendo (10,5 1,5 0 -1 0 0 0 0).

Los valores obtenidos son los coeficientes de Haar de esa lista. Podríamos obtener la lista como la suma de los productos de cada coeficiente por su función de Haar, es decir, $f = \sum_{k=1}^n c_k h_k$.

Realicemos el cálculo para nuestra lista ejemplo:

$$\begin{aligned}
 (12 12 12 12 8 8 10 10) &= 10,5h_1 + 1,5h_2 - h_4 = \\
 &= 10,5(1 1 1 1 1 1 1 1) + 1,5(1 1 1 1 -1 -1 -1 -1) \\
 &\quad - (0 0 0 0 1 1 -1 -1) = \\
 &= (12 12 12 12 9 9 9 9) - (0 0 0 0 1 1 -1 -1).
 \end{aligned}$$

Efectivamente obtenemos la lista original. Más adelante tendremos que coger los coeficientes que más información nos aporten. Nos fijamos en que si hubiésemos quitado el coeficiente menos importante (distinto de cero), hubiera sido el -1 , y entonces la aproximación de la lista inicial hubiera sido $(12\ 12\ 12\ 12\ 9\ 9\ 9\ 9)$ la cual refleja bastante bien la original.

Para realizar este proceso la lista inicial debe de tener tamaño 2^n con n un número natural y el número de iteraciones, dado que la lista se divide a la mitad en cada paso, es n . En el ejemplo mostrado $n = 3$.

Vamos a examinar la búsqueda de un proceso inverso a la transformada. Es claro que si el cálculo es:

$$\text{avg} = \frac{x + y}{2}, \quad \text{dif} = \frac{x - y}{2},$$

entonces su inverso es:

$$x = \text{avg} + \text{dif}, \quad y = \text{avg} - \text{dif}.$$

Mediante este razonamiento se puede construir un procedimiento iterativo análogo al explicado que calcule la transformada inversa. Ejemplificamos esto sobre la lista de coeficientes del ejemplo $(10,5\ 1,5\ 0\ -1\ 0\ 0\ 0\ 0)$:

$$(12\ 9\ 0\ -1\ 0\ 0\ 0\ 0) \rightarrow (12\ 12\ 8\ 10\ 0\ 0\ 0\ 0) \rightarrow (12\ 12\ 12\ 12\ 8\ 8\ 10\ 10).$$

Sin embargo, un detalle fundamental de la transformada explicada es que los coeficientes c_k de la base de Haar no son totalmente objetivos, algunos son más importantes que otros. El motivo de este hecho es que no hemos usado la base de Haar normalizada. Matemáticamente:

$$h_k^p = \frac{h_k}{\|h_k\|_p} \implies f = \sum_{k=1}^n c_k h_k = \sum_{k=1}^n c_k \|h_k\|_p h_k^p = \sum_{k=1}^n c_k^p h_k^p$$

donde h_k^p son las funciones del sistema de Haar normalizado en $L_p[0, 1]$ ($1 < p < \infty$) y $c_k^p = c_k \|h_k\|_p$ los coeficientes sobre el sistema normalizado. Los nuevos coeficientes c_k^p reflejan mucho mejor el valor de los píxeles y son los adecuados a la hora de pasar cualquier algoritmo por la lista. Tenemos derecho a usar el sistema de Haar normalizado en $L_p[0, 1]$ para $1 < p < \infty$ porque en la Proposición 2.4.10 se ha probado que es una base greedy.

La norma de la función de Haar h_n es:

$$\|h_n\|_p = \frac{1}{2^{\frac{k}{p}}} \text{ donde } n = 2^k + s, \quad k = 0, 1, 2, \dots \text{ y } s = 1, 2, \dots, 2^k.$$

Los coeficientes de la lista ejemplo respecto del sistema de Haar normalizado en $L_2[0, 1]$ son $(10,5\ 1,5\ 0\ -0,7\ 0\ 0\ 0\ 0)$, en donde se ha redondeado a 2 decimales.

Antes de continuar es interesante calcular la complejidad de esta transformación sobre una lista, el número de operaciones aritméticas necesarias en función del tamaño de los datos. En nuestro ejemplo en el primer paso hicimos 16 operaciones (4 sumas, 4 restas y 8 divisiones). Siguiendo este razonamiento el número total de operaciones de todas las iteraciones es $16 + 8 + 4 = 28$.

Para una lista de tamaño $N = 2^n$ el total de operaciones es:

$$\sum_{i=2}^{n+1} 2^i = \sum_{i=0}^{n+1} 2^i - 3 = \frac{1 - 2^{n+2}}{1 - 2} - 3 = 2^{n+2} - 4 = 4(N - 1).$$

La complejidad es de orden lineal y por tanto es baja. Comprobamos efectivamente que en el ejemplo $4(8) - 4 = 28$. La complejidad de la inversa de la transformada es similar. Hay que considerar que ya no se realiza la división entre 2. Veamos:

$$\sum_{i=1}^n 2^i = \sum_{i=0}^n 2^i - 1 = \frac{1 - 2^{n+1}}{1 - 2} - 1 = 2^{n+1} - 2 = 2(N - 1),$$

también de orden lineal y algo más baja que la de la transformada.

3.1.3. Normalización de la transformada en $L_p[0, 1]$

Acabamos de introducir la noción de sistema de Haar normalizado y el uso de coeficientes sobre este sistema. Claramente una opción para obtener los coeficientes normalizados c_k^p es calcular los coeficientes sin normalizar c_k y posteriormente ajustar multiplicando cada uno por su norma en $L_p[0, 1]$ ($1 < p < \infty$), la cual denotamos por $\|h_k\|_p$.

No obstante, surge la pregunta de si existe un procedimiento iterativo análogo al ya explicado de manera que se obtengan directamente los coeficientes normalizados. En otras palabras, queremos saber si en vez de hacer medias y diferencias entre dos hacemos otra operación es posible conseguir los normalizados.

En caso afirmativo, ¿qué procedimiento es más eficiente? Esta pregunta es fundamental para saber si solamente hemos resuelto una curiosidad con cierto interés matemático e informático o hemos construido un algoritmo que mejora la idea inicial. En este apartado vamos a dar respuesta a ambas preguntas.

En primer lugar vamos a abstraer el procedimiento sin normalizar para una lista general $(x_1 \ x_2 \ \dots \ x_{N-1} \ x_N)$ donde $N = 2^n$ y $n \in \mathbb{N}$. Sabemos que para una lista de tamaño 2^n la transformada usa n iteraciones, las cuales sobre esta lista son:

$$\left(\frac{x_1 + x_2}{2} \ \dots \ \frac{x_{N-1} - x_N}{2} \right) \xrightarrow{n \text{ iter.}} \left(\underbrace{\frac{x_1 + \dots + x_N}{2^n}}_{c_1} \ \dots \ \underbrace{\frac{x_{N-1} - x_N}{2}}_{c_N} \right)$$

Se han señalado los coeficientes c_1 y c_N . Más adelante nos resultará útil disponer de la expresión de cualquier c_j con $j \in \{1, \dots, N\}$. Atendiendo al procedimiento ésta es

$$c_j = \frac{n_j}{2^{n-k}} \text{ donde } j = 2^k + s \text{ con } k = 0, 1, \dots \text{ y } s = 1, 2, \dots, 2^k.$$

La expresión del numerador no es relevante para la prueba que se pretende realizar. Atendiendo a la relación $c_j^p = c_j \|h_j\|_p$ las modificaciones sobre el proceso son las siguientes:

- 1) Hay que realizar una normalización uniforme sobre los elementos de la lista original. Dicha normalización consiste en dividir todos los coeficientes por $2^{\frac{n}{p}}$. Notemos que es indiferente realizar esto al principio sobre los elementos de la lista que al final sobre los coeficientes obtenidos. Como la división hace más pequeños los coeficientes ($n \gg 0$) computacionalmente es mejor realizarla al final para reducir los errores de redondeo del ordenador.
- 2) En cada iteración en vez de dividir por 2 se hará por $2^{\frac{1}{q}}$ donde q es aquel que verifica $\frac{1}{p} + \frac{1}{q} = 1$, relación bien conocida en Análisis Funcional.

Teniendo en cuenta las dos modificaciones, el nuevo procedimiento es:

$$\left(\frac{x_1 + x_2}{2^{\frac{1}{q}}} \dots \frac{x_{N-1} - x_N}{2^{\frac{1}{q}}} \right) \xrightarrow{n \text{ iter.}} \left(\underbrace{\frac{x_1 + \dots + x_N}{2^{\frac{n}{q}} 2^{\frac{n}{p}}}}_{c_1^p} \dots \underbrace{\frac{x_{N-1} - x_N}{2^{\frac{1}{q}} 2^{\frac{n}{p}}}}_{c_N^p} \right) \quad (3.1)$$

Nuevamente es bueno saber las expresiones de c_1^p y c_N^p y podríamos verificar si cumplen la relación que estamos buscando. Esto no probaría nada porque podría ser que sí la verificasen pero no el resto de coeficientes intermedios. Por eso vamos a hallar la expresión de c_j^p para cualquier $j \in \{1, \dots, N\}$.

$$c_j^p = \frac{n_j}{2^{\frac{n-k}{q}} 2^{\frac{n}{p}}} \text{ donde } j = 2^k + s \text{ con } k = 0, 1, \dots \text{ y } s = 1, 2, \dots, 2^k.$$

Obsérvese que los n_j son exactamente los mismos que en el caso sin normalizar. Claramente c_1^p y c_N^p verifican la expresión pues si $j = 1$ entonces $k = 0$ y si $j = N = 2^n$ entonces $k = n - 1$.

Después de explicar las modificaciones necesarias para realizar una posible construcción iterativa que proporcione los coeficientes normalizados vamos a demostrar la veracidad de dicha construcción. Para ello basta probar $c_j^p = c_j \|h_j\|_p$ para todo $j = 1, \dots, N$. Podemos suponer que los numeradores $n_j \neq 0$ ya que si son cero la igualdad es cierta trivialmente. El primer paso de la prueba es

$$\frac{n_j}{2^{\frac{n-k}{q}} 2^{\frac{n}{p}}} = \frac{n_j}{2^{n-k+\frac{k}{p}}} \iff \frac{n-k}{q} + \frac{n}{p} = n - k + \frac{k}{p},$$

ahora multiplicamos en ambos miembros por $p q$

$$(n - k) p + n q = (n - k) p q + k q.$$

Como $\frac{1}{p} + \frac{1}{q} = 1$ entonces p y q cumplen que su suma es igual a su producto $p + q = p q$. Usamos este hecho para simplificar:

$$n q = (n - k) q + k q,$$

y como $q \neq 0$ por ser $1 < p < \infty$ podemos dividir por él o simplemente sacando factor común llegamos a $n = n - k + k$ como queríamos demostrar. Podemos afirmar ahora que la construcción explicada en 3.1 proporciona los coeficientes normalizados.

Después de dar respuesta a la primera pregunta vamos responder la segunda. Para ello hemos de observar que la complejidad de este procedimiento es $5N - 4$ donde N es el tamaño de la lista. En esa última iteración sobre los N elementos se está multiplicando por un valor constante $2^{\frac{n}{p}}$. De la otra forma para cada elemento hay que calcular su norma y eso es más costoso. Por tanto, hemos ganado con esta construcción.

Respecto a la transformada inversa, es necesario puntualizar los cambios derivados de estas modificaciones:

- 1) El normalizado uniforme se anula multiplicando por $2^{\frac{n}{p}}$ y se hará principio para obtener coeficientes más grandes y mejorar la precisión del ordenador.
- 2) En el procedimiento iterativo si el calculo es:

$$\text{avg} = \frac{x + y}{2^{\frac{1}{q}}}, \quad \text{dif} = \frac{x - y}{2^{\frac{1}{q}}},$$

entonces su inverso es:

$$x = \frac{\text{avg} + \text{dif}}{2^{\frac{1}{p}}}, \quad y = \frac{\text{avg} - \text{dif}}{2^{\frac{1}{p}}}. \quad (3.2)$$

De este modo cuando los coeficientes sean los normalizados el número de operaciones necesarias para la inversa coincide con el de la transformada, siendo $5N - 4$.

Nota. La normalización más común y que usaremos en el desarrollo del trabajo es la de $L_2[0, 1]$. Este espacio es de Hilbert y tenemos la igualdad $p = q = 2$ por lo que los cálculos se simplifican. Se divide por $\sqrt{2}$ en cada iteración de la transformada (en la inversa también) y en la normalización uniforme se divide (y en la inversa se multiplica) por $(\sqrt{2})^n$.

3.1.4. Implementación de la transformada de Haar discreta

La transformada de Haar discreta va a ser la herramienta fundamental del trabajo informático. Ahora vamos a exponer la implementación realizada. En primer lugar explicaremos la obtención de los coeficientes y luego las consideraciones necesarias para realizar

el cálculo de la transformada inversa. Por último hablaremos de la aproximación greedy y los métodos implementados para la elección de los mejores coeficientes.

Obtención de los coeficientes

Las explicaciones del cálculo de los coeficientes ya están debidamente detalladas a lo largo de esta Sección 3.1 por lo tanto se va a mostrar el pseudo-código y posteriormente se comentarán breves detalles.

Algorithm 1 Transformada de Haar de una lista

```

procedure HAAR_TRANSFORM(list, p)
    if p == 0 then
        val  $\leftarrow 2$ 
    else
        val  $\leftarrow 2^{\frac{p-1}{p}}$ 
    haar  $\leftarrow$  haar_split(list, [ ], val)
    if p  $\neq 0$  then ▷ Normalizado uniforme
        for i  $\leftarrow 0$  to len(haar) do
            haar[i]  $\leftarrow$  haar[i] * len(haar) $^{\frac{1}{p}}$ 
    return list+offset
```

Las entradas son la lista sobre la calcular la transformada y *p* que indica el espacio en donde se va a normalizar. En caso de *p* sea cero se calculan los coeficientes sin normalizar. Obsérvese que a `haar_split` se le podría pasar el parámetro *p* en vez del valor *val* pero entonces en cada iteración se tendría que calcular *val* aumentando la complejidad.

Algorithm 2 Particionamiento para Haar recursivo

```

procedure HAAR_SPLIT(list, offset, value)
    if len(list)  $\geq 2$  then
        avgs  $\leftarrow$  [ ], difs  $\leftarrow$  [ ]
        for i  $\leftarrow 0$  to len(list) step 2 do
            avgs[i]  $\leftarrow$  (list[i] + list[i + 1])/value
            difs[i]  $\leftarrow$  (list[i] - list[i + 1])/value
        return haar_split(avgs, difs+offset, value) ▷ Concatenación de listas
    else
        return list+offset
```

Era de imaginar que fuese una función recursiva y en éstas la condición de parada es clave. Mientras la lista tenga tamaño mayor o igual que 2 seguiremos calculando transfor-

madas.

El procedimiento consiste en realizar las sumas y diferencias divididas por el valor correspondiente para la normalización. Después, sólo hay que volver a repetir el proceso dejando fijada la segunda parte, la cual se denomina `offset`, y volver a realizar el procedimiento sobre la primera parte. En la última iteración se devuelve la concatenación de `list` y todos los `offset` acumulados hasta el momento. Se ha usado el operador `+` que sobre listas denota concatenación.

Restauración de los coeficientes

Vamos a examinar la búsqueda de un proceso inverso a la transformada. Lo primero es deshacer el normalizado uniforme multiplicando la lista entera por $2^{\frac{n}{p}}$ (2^n es la longitud de la lista). Posteriormente se calcula el valor que se usa para dividir, $val = 2^{\frac{1}{p}}$, y se llama a la función de particionamiento `reverse_split`.

Algorithm 3 Transformada inversa de Haar de una lista

```
procedure REVERSE_HAAR(list, offset)
    if p == 0 then
        val  $\leftarrow$  1
    else
        val  $\leftarrow 2^{\frac{1}{p}}
    for i  $\leftarrow$  0 to len(list) do
        list[i]  $\leftarrow$  list[i] * len(list) $^{\frac{1}{p}}$ 
    reverse  $\leftarrow$  reverse_split(list[: 1], list[1 :], 0, val)
    return reverse$ 
```

Igual que antes si $p = 0$ entonces los coeficientes no están normalizados y se contemplan las modificaciones necesarias. Vamos a mostrar y explicar la última función `reverse_split`.

A partir de la ecuación 3.2 sólo debemos preocuparnos por la condición de parada de nuestro algoritmo recursivo y por cómo particionar la lista de manera opuesta a como lo hacíamos en `haar_split`. Estos dos hechos están relacionados. Veamos:

En la primera iteración el elemento libre es el primero y el resto de la lista está fija. Se calcula la suma y diferencia de ese elemento dividido por *val* con el primer elemento del resto de la lista, en el pseudo-código denominada `the_rest`. En la siguiente iteración aumentamos el exponente `power` y hacemos lo mismo fragmentando la primera parte con 2^{power} elementos y la segunda desde ahí hasta el final (notación $[2^{power} :]$). La condición de parada es que la primera partición de la lista sea más grande que la segunda.

Algorithm 4 Particionamiento inverso para Haar recursivo

```

procedure REVERSE_SPLIT(front, the_rest, power, value)
    reverse  $\leftarrow$  []
    for i  $\leftarrow$  0 to len(front) do
        reverse.insert((front[i] + the_rest[i])/value)
        reverse.insert((front[i] - the_rest[i])/value)
    if len(the_rest) > len(reverse) then
        the_rest = the_rest[ $2^{power} :$ ]
        power  $\leftarrow$  power + 1
        return reverse_split(reverse, the_rest, power, value)
    else
        return reverse

```

Todas las aplicaciones que se van a construir están basadas en esta teoría y en particular en estas funciones. Son el cuello de botella del proyecto ya que todo pasa por ellas, por eso se han implementado contemplando todos los casos y de la manera más eficiente posible.

Aproximación greedy

Un aspecto fundamental de la aproximación es saber el nivel de detalle y precisión que se desea en la aproximación. La compresión que se va a construir puede ser con pérdidas o sin pérdidas. Nos vamos a centrar especialmente en la que admite pérdida. En ese caso cuanto menos error se quiera cometer más coeficientes es necesario almacenar y menos compresión se realiza y viceversa.

Existen diversas maneras de determinar los coeficientes que van a ser descartados. Aquí es muy importante que los coeficientes que hayamos obtenido sean respecto de la base normalizada para tratar equivalentemente a todos. Las normas de las funciones del sistema de Haar asociadas a los primeros elementos de la lista serán grandes (como mucho uno) y conforme nos alejemos serán más pequeñas.

A continuación se exponen los algoritmos desarrollados para descartar los coeficientes:

- El primer algoritmo para determinar los coeficientes a ser descartados que hemos implementado es **thresholding**, un algoritmo umbral. Se elige un valor umbral $\varepsilon > 0$ que consideremos adecuado a la imagen y recorremos todos los coeficientes borrando aquellos que cumplan $|\hat{c}_i| < \varepsilon$. Del mismo modo se puede llevar un contador de cuantos coeficientes son descartados y fácilmente calculamos el porcentaje de

descartados. Elegir $\varepsilon = 0$ es usar una compresión sin pérdidas.

El algoritmo es bastante sencillo y eficiente pues solo recorre la lista una vez y en cada elemento sólo hace una comprobación y en algunos casos una asignación. Debido a la sencillez no se va mostrar el pseudo-código.

- El segundo algoritmo que se ha llevado a cabo es `m_term`. Como su nombre indica es un algoritmo de m -términos, es decir, calcula los m mejores coeficientes y el resto los descarta. La implementación de este procedimiento se ha realizado calculando una lista de tamaño m en donde almacenamos los mejores $|\hat{c}_i|$ hasta el momento. Descartamos el resto usando `thresholding` con $\varepsilon = \min_m\{d_1, \dots, d_m\}$ donde d_1, \dots, d_m son los coeficientes de la lista.

Ésta es de las pocas implementaciones eficientes pero tiene una desventaja y es que cuando haya varios valores iguales el `thresholding` los borraría todos y no estaríamos quedándonos exclusivamente con m términos.

Después de establecer la aproximación greedy es necesario llevar a cabo una compresión sobre ellos. Muchos de estos coeficientes son ceros por lo que la compresión va a ser satisfactoria y en breve explicamos la posibilidad elegida en este caso. Posteriormente la compresión se enviaría al receptor, quien lo descomprimiría y revertiría la transformada de Haar obteniendo una aproximación de la función de partida.

Una herramienta útil para saber cómo de buena es la reconstrucción de la función sería tener una medida del error medio. Hay varias opciones como usar el error cuadrático medio MSE aunque aquí se ha implementado una función `error` que calcula la media de las diferencias (en valor absoluto) entre todos los valores de la lista original y la revertida. Si la lista a tratar es de tamaño N y denotamos como $A = (a_i)_{i=1}^N$ a la original y $\hat{A} = (\hat{a}_i)_{i=1}^N$ a la revertida entonces el error es:

$$\text{error} = \frac{1}{N} \sum_{i=1}^N |a_i - \hat{a}_i|.$$

3.1.5. Compresión de los coeficientes

El objetivo de lo que vamos a realizar es facilitar y mejorar la compresión de información así que debemos de definir dos conceptos que controlan la medida de la compresión que se realiza. Dichos conceptos son el *ratio de compresión*

$$\text{ratio de compresión} = \frac{\text{Tamaño del archivo de salida}}{\text{Tamaño del archivo de entrada}},$$

y su inverso, el *factor de compresión*:

$$\text{factor de compresión} = \frac{\text{Tamaño del archivo de entrada}}{\text{Tamaño del archivo de salida}}.$$

La compresión de información la vamos a realizar usando como base una compresión de datos sin pérdidas como es la *codificación de longitud* (RLE, *run-length encoding*) [12].

La idea detrás de este enfoque para la compresión de datos es la siguiente: si un elemento d aparece n veces consecutivas en la secuencia de entrada, reemplazamos las n ocurrencias con el par único nd . De esta manera comprimiríamos la cadena $BBBBBGNN$ a $5BG2N$.

La dificultad de este procedimiento es saber si un elemento contiene un dato como puede ser B o una cantidad como 5 . Hay diversas soluciones para esto, lo más importante es tener en cuenta los objetos que pretendemos comprimir para hacer mejoras que un modelo general no tendría.

En nuestro caso necesitamos este procedimiento para comprimir los coeficientes greedy. Ese es el momento exacto donde se realiza la compresión y el envío de la información al receptor. En particular, no sabemos en qué rango están los valores. Ahora bien, sabemos que va a haber muchos elementos consecutivos que sean ceros y que es muy raro que haya varios elementos consecutivos con el mismo valor a excepción de que ese valor sea cero. Podemos limitar la codificación al valor cero. Esto supone una ventaja porque no necesitamos almacenar el valor que se repite, ya sabemos que es cero.

La solución para distinguir los valores que indican una cantidad es usando un valor centinela. Debido a que no sabemos el rango de los datos la idea se ha llevado a cabo es elegir la parte entera del máximo de los valores más uno. Así ese valor es único y los consideraremos el centinela. Lo guardaremos ya que es necesario para la descompresión.

La lista $[3,5\ 2\ 0\ 0\ 0\ 1]$ sería comprimida a $[3,5\ 2\ \underline{4}\ 3\ 1]$. Tal y como vemos cuando aparezca el valor centinela, el siguiente valor es una cantidad de ceros.

3.2. Transformada de Haar para aproximación de señales

En esta sección se propone la primera aplicación de aproximación greedy mediante la transformada de Haar discreta. Los objetos que van a ser aproximados y comprimidos son señales. Comenzaremos realizando un diseño de la solución en donde se discute bajo qué condiciones de la señal podemos asegurar éxito en la aproximación. Más adelante se exponen los aspectos más importantes de los algoritmos. En último lugar un análisis exhaustivo de los resultados obtenidos y la optimización de diferentes parámetros del problema.

El código de esta sección está en la carpeta `signal_code` a la que se puede acceder a través de https://github.com/Carlossamu7/TFG/tree/master/signal_code.

El estudio habitual de transformadas wavelet sobre señales es para el caso continuo [14]. La novedad de esta propuesta es que es de manera discreta lo cual implica una mejor eficiencia en tiempo a la vez que una aproximación satisfactoria. Familiarizados con la base de conocimiento necesaria para abordar esta aplicación cabe formularse algunas preguntas. Estas preguntas surgen de manera natural con el planteamiento del problema y durante el desarrollo de la sección se pretende responderlas:

- (S1) Al discretizar una señal para comenzar la aproximación ya se comete un error inicial.
¿Qué cantidad de información se pierde en esta discretización?
- (S2) ¿Los coeficientes greedy de una señal tienen algún significado?
- (S3) El umbral que se usa en el algoritmo de aproximación está ligado al error que se comete. ¿Se puede optimizar la relación umbral-error?
- (S4) El número de trozos en los que se discretiza la señal, N , también influye. ¿Se puede optimizar la relación N -error?
- (S5) La normalización de los coeficientes que usamos es en $L_2[0, 1]$. ¿Cuál es la mejor elección de p para la normalización?

3.2.1. Diseño de la solución

El primer paso es conocer que las señales que vamos a manejar son de dos formas. La primera es que no se conozca una expresión asociada y sólo se posea un muestreado de la señal. El segundo caso es que conozcamos su expresión analítica. Para el primer caso ya se puede entrar en acción y usar los algoritmos implementados. Para el segundo es necesario analizar la escena.

Sea f la función de la señal. Se toman 2^n puntos equiespaciados en un cierto dominio y se evalúa f en esos puntos. En este contexto entra en juego la pregunta (S1). La respuesta es que sí se puede medir el error cometido sólo en el hecho de aproximar f por una función simple. Necesitamos un concepto más restrictivo que el de continuidad, vamos a exigir que las señales sean lipschitzianas.

Se dice que una función $f : A \rightarrow \mathbb{R}$, $A \subset \mathbb{R}$ es *lipschitziana* cuando existe una constante $M \geq 0$ verificando que:

$$|f(t) - f(s)| \leq M |t - s| \quad \forall t, s \in A. \quad (3.3)$$

Existe una constante mínima $M_0 \geq 0$ que verifica la desigualdad 3.3,

$$M_0 = \sup \left\{ \frac{|f(t) - f(s)|}{|t - s|} : t, s \in A, t \neq s \right\}$$

y se dice que M_0 es la *constante de lipschitz* de f , en honor al matemático alemán Rudolph Lipschitz (1832-1903).

Sea I un intervalo (no trivial) de \mathbb{R} , $f : I \rightarrow \mathbb{R}$ una señal que es lipschitzana y $n \in \mathbb{N}$. Si tomamos 2^n puntos equiespaciados en el intervalo I y la longitud del intervalo es L entonces en virtud de 3.3 tenemos que

$$|t - s| \leq \frac{L}{2^n} \implies |f(t) - f(s)| \leq \frac{ML}{2^n}.$$

De esta forma el error máximo cometido en cada punto es $ML/2^n$. El único dato que puede ser difícil de calcular para ciertas señales es M . Lo idóneo siempre que sea posible es trabajar con la mejor constante, la constante de lipschitz, aunque cuando no sea posible trabajaremos con otra.

Una caracterización que permite hallar dicha constante es la siguiente. Sea I un intervalo (no trivial) de \mathbb{R} y $f \in \mathcal{C}(I) \cap \mathcal{D}(I^o)$. Las siguientes afirmaciones son equivalentes:

- (I) f es lipschitziana.
- (II) f' está acotada, es decir, existe $M \geq 0$ tal que $|f'(x)| \leq M$ para todo $x \in I^o$.

En caso de que se verifiquen (I) y (II), la constante de lipschitz de f viene dada por

$$M_0 = \sup\{|f'(x)| : x \in I^o\}.$$

Esta caracterización es muy útil para conocer a priori el error máximo cometido al discretizar la señal.

3.2.2. Algoritmos

La base de los algoritmos necesarios para esta tarea ya los tenemos. Se va a explicar brevemente el diseño realizado para la compresión de la información, el envío al receptor y la descompresión. El Algoritmo 5 contiene un esquema en pseudo-código del diseño.

Se añaden breves explicaciones de este algoritmo:

- Se utilizan las entradas `signal_f` y `N` para construir N puntos equiespaciados en el intervalo `dom` y las imágenes mediante `signal_f` de dichos puntos. En caso de que no exista $n \in \mathbb{N}$ tal que $N = 2^n$ se extiende la señal por ceros a la más pequeña n tal que $N < 2^n$.
- Se calcula la transformada de Haar mediante el Algoritmo 1. Después de esto se realiza la aproximación greedy descartando los coeficientes normalizados \hat{c}_i . El parámetro `fun` indica el método de aproximación. Se dispone de los dos métodos ya explicados: (1) `thresholding` que descarta aquellos que verifican la condición $|\hat{c}_i| < \varepsilon$ y (2) `m_term` que escoge los m mejores términos.
- Se comprime la imagen usando la compresión sin pérdidas codificación de longitud.

Algorithm 5 Transformada de Haar para señales

```

procedure EXPERIMENT(signal_f, dom, N, fun, param)
    puntos  $\leftarrow N$  puntos equiespaciados en dom
    signal  $\leftarrow \text{signal\_f}(\text{puntos})
    ext  $\leftarrow \text{extend\_signal}(\text{signal})                                 $\triangleright$  Sólo extiende si es necesario
    haar_signal  $\leftarrow \text{haar\_transform}(\text{ext}, [ ])
    greedy_signal, perc, ratio  $\leftarrow \text{fun}(\text{haar\_signal}, \text{param})
    comp_signal, cent  $\leftarrow \text{compress\_signal}(\text{greedy\_signal})
    — ENVÍO DE comp_signal AL RECEPTOR —
    uncomp_signal  $\leftarrow \text{uncompress\_signal}(\text{comp\_signal}, \text{cent})
    rev_signal  $\leftarrow \text{reverse\_haar}(\text{uncomp\_signal}[ : 1], \text{uncomp\_signal}[1 :], 0)
    rev_signal  $\leftarrow \text{crop\_size}(\text{rev\_signal}, \text{N})                       $\triangleright$  Sólo recorta si se extendió
    err  $\leftarrow \text{error}(\text{signal}, \text{rev\_signal})
    factor  $\leftarrow \text{diff\_size}(\text{signal}, \text{rev\_signal})
    return comp_signal, perc, ratio, err, factor$$$$$$$$$$ 
```

En este instante se envía el objeto comprimido y el valor centinela al receptor, quien continuará este procedimiento para obtener la información original.

- El receptor descomprime la imagen para lo cual es necesario el centinela.
- Se restaura la señal original usando la transformada de Haar inversa correspondiente al Algoritmo 3. En caso de que la señal fuese extendida ahora se recorta al tamaño de la señal original completando con éxito este proceso.
- Como esto es un análisis del procedimiento se hallan en último lugar dos medidas para controlar la calidad de la compresión: el error medio y el factor de compresión.

3.2.3. Resultados y análisis

Vamos a elegir algunas señales para observar el resultado de los algoritmos implementados. Las señales elegidas son $f(x) = \sin(x)$ y $g(x) = x \sin(x) + x \cos(x)$. El dominio sobre el cual las vamos a aproximar es $[0, 2\pi]$ y el número de trozos $N = 512$.

Antes de continuar vamos probar que ambas son lipschitzianas, vamos a calcular una constante de lipschitz, o al menos una que acote la derivada y finalmente calcular el error máximo por punto cometido en la discretización.

Comenzamos con la señal $f(x) = \sin(x)$:

$$|f'(x)| = |\cos(x)| \leq 1$$

por lo que

$$\text{error}(f) \leq \frac{1 \cdot 2\pi}{2^9} \approx 0,012272.$$

El caso de g no es tan directo. Veamos:

$$\begin{aligned} |g'(x)| &= |\sin(x) + \cos(x) + x(-\sin(x) + \cos(x))| \\ &\leq |\sin(x) + \cos(x)| + |x| |\sin(x) + \cos(x)| \\ &\leq \sqrt{2} + 2\pi\sqrt{2} = (2\pi + 1)\sqrt{2}. \end{aligned}$$

Nótese que no tenemos certeza de que esa sea la constante de Lipschitz, pero la acotación es buena y podemos usarla. Así el error máximo por punto es:

$$\text{error}(g) \leq \frac{(2\pi + 1)\sqrt{2} \cdot 2\pi}{2^9} \approx 0,1264.$$

Estudiaremos el efecto que tienen tres umbrales ε distintos para el **thresholding**. Comenzamos con $\varepsilon = 0,005$.

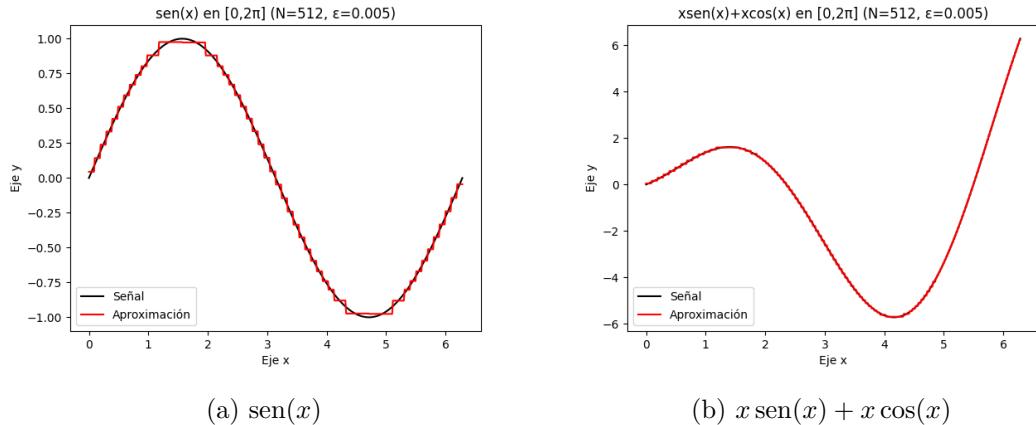


Imagen 3.2: Señales en $[0, 2\pi]$ para $N = 512$ después de **thresholding** con $\varepsilon = 0,005$

En negro encontramos la señal original y en rojo la aproximación. Claramente el nivel de detalle en la aproximación de 3.2b es mayor que el de 3.2a. Esto se debe a que 3.2b tiene mucho más cambio por lo que las diferencias son, en valor absoluto, mayores y por tanto a pesar de usar el mismo umbral en ambas en 3.2a se descartan más coeficientes. Efectivamente en la del seno el porcentaje de descarte es 91,21 mientras que en la otra 74,8.

Los errores medios cometidos son bajos, el de la Imagen 3.2a es 0,0209 y el de la Imagen 3.2b es 0,0365. Para poder observar el efecto del umbral en la precisión de la aproximación vamos a cambiar el umbral a uno mayor, como $\varepsilon = 0,02$.

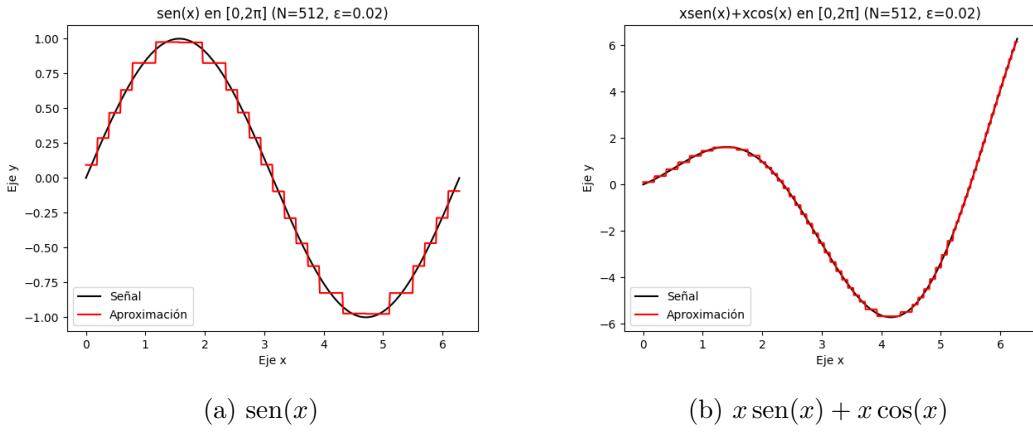


Imagen 3.3: Señales en $[0, 2\pi]$ para $N = 512$ después de **thresholding** con $\varepsilon = 0,02$

Con este segundo umbral los trozos de la aproximación son mayores. Nuevamente la aproximación de 3.3b es mejor. Los porcentajes de descarte son respectivamente 95,9 y 88,48 y los errores medios 0,0408 y 0,0783. Es lógico que a mayores umbrales se descarte un mayor número de coeficientes aumentando consiguientemente el error.

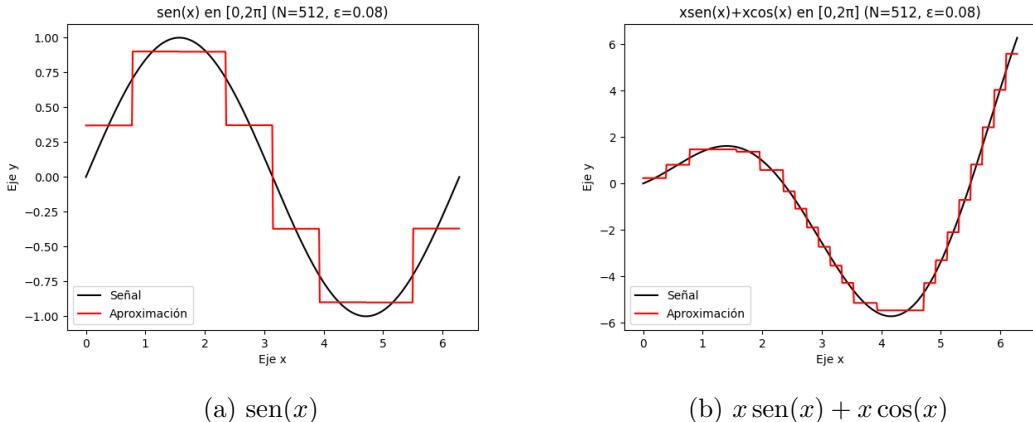


Imagen 3.4: Señales en $[0, 2\pi]$ para $N = 512$ después de **thresholding** con $\varepsilon = 0,08$

Por último un umbral $\varepsilon = 0,08$. Los escalones de la gráfica aproximada son grandes ofreciendo una aproximación más gruesa de la señal. Concluimos que el umbral es el elemento que nos permite escoger el nivel de detalle de la aproximación. Estos últimos porcentajes de descarte son 99,02 y 95,9 y los errores medios son 0,128 y 0,2257.

Para hacer un análisis conjunto vamos a recopilar en la siguiente tabla todos los resultados obtenidos junto con los factores de compresión, de los cuales aún no hemos comentado nada aún.

Tabla 3.1: Recopilación resultados de las señales

Ejemplo	Umbral	Porcentaje de descarte (%)	Error medio	Factor de compresión
$\sin(x)$	0,005	91,21	0,0209	8,67797
$\sin(x)$	0,02	95,9	0,0408	16,5161
$\sin(x)$	0,08	99,02	0,128	46,5455
$x \sin(x) + x \cos(x)$	0,005	74,8	0,0365	3,58042
$x \sin(x) + x \cos(x)$	0,02	88,48	0,0783	7,21127
$x \sin(x) + x \cos(x)$	0,08	95,9	0,2257	16,5161

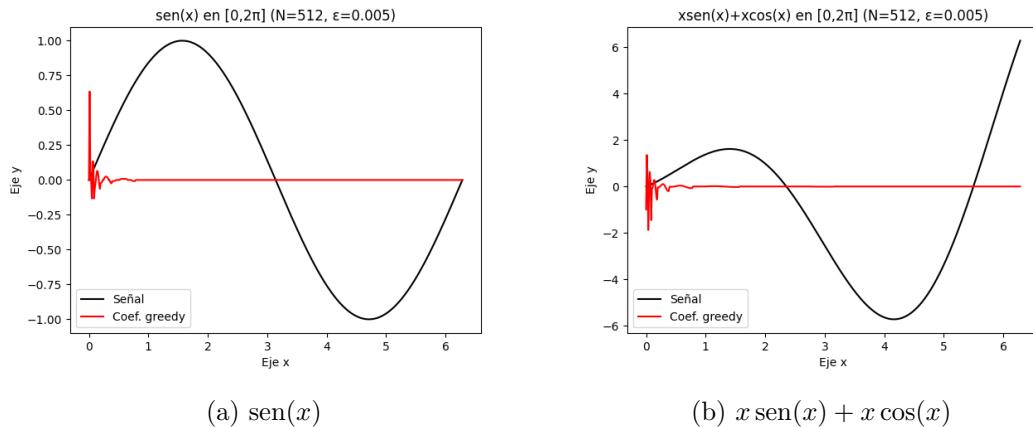
En el caso del seno los porcentajes de descarte están son mayores al 90 % y en la otra señal hay más variabilidad a pesar de usar los mismos umbrales. Los errores medios del seno son más bajos y además tenían un error máximo de discretización inferior.

Respecto a los factores de compresión son bastante buenos y permitirían almacenar en poca memoria una señal que puede ser muy importante. El mejor factor es 46,5455 para el seno con umbral 0,08. Es normal que sea alto pues sólo se conserva un 1 % de los coeficientes.

Señales de los coeficientes respecto de la base de Haar

Es lógico pensar que los coeficientes calculados por la transformada de Haar puedan tener algún significado. La pregunta (S2) que nos formulábamos surgió de ésta idea.

Para ello lo primero es pintar la señal de los coeficientes para los dos señales que tenemos como ejemplo.

Imagen 3.5: Señales de los coeficientes greedy después de thresholding con $\varepsilon = 0,005$

Aparentemente no hay un significado explícito del resultado. Es normal que al principio haya más onda en la señal de los coeficientes y al final sea más plana. Esto se debe a la construcción en la que vamos dejando fijas las partes derechas de las particiones de las listas. Además los primeros elementos de la lista tienen una norma mayor a los últimos.

Una situación que alguien puede pensar que se puede dar es la siguiente. La mitad derecha de la señal son las diferencias de todas las parejas de la lista divididas por $\sqrt{2}$, ya que se está normalizando en $L_2[0, 1]$. Por otro lado la definición de derivada es

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

Hay una similitud entre esas diferencias y esta definición. De hecho $h = L/2^n$, pero en el denominador estamos dividiendo por $\sqrt{2}$ un número que no tiene nada que ver con $h = L/2^n$ y que distorsiona la discretización de la derivada acabando con esta teoría. Por tanto, para estas señales no hay un significado gráfico aparentemente explícito de los coeficientes.

3.2.4. Optimización del error medio y la compresión

En la aproximación realizada hay dos parámetros que están en juego y afectan directamente al error medio. El primero de ellos es el umbral ε . La cuestión (S3) es precisamente esto. Valores pequeños de ε conducen a errores medios pequeños y valores grandes hacen que un número alto de coeficientes se descarten aumentando el error medio.

El segundo parámetro es el número de trozos en el cual se discretiza la señal $N = 2^n$, $n \in \mathbb{N}$. De ahí ya tenemos un error inicial en el hecho de discretizarla a lo cual hay que sumarle el error producido por la aproximación bajo esa elección de N . Valores de N muy grandes hacen que ambos errores sean muy pequeños pero conlleva la necesidad de disponer de mayor memoria y mayor número de iteraciones en el algoritmo aumentando la complejidad. Ésta es la pregunta (S4).

Además surge la necesidad de cuestionarse por qué ceñirse a $L_2[0, 1]$ cuando el trabajo matemático dice que la base de Haar es greedy en $L_p[0, 1]$ con $1 < p < \infty$. ¿Es $p=2$ la mejor elección? Sería muy interesante aprovechar ese resultado matemático para informáticamente analizar si normalizando en distintos espacios se puede conseguir un error mejor. De este modo tenemos una pregunta que necesita de todo el desarrollo del trabajo para conseguir una potencia computacional. Esta pregunta se corresponde a (S5).

En este apartado vamos a estudiar posibles optimizaciones de las relaciones de dichos parámetros con el error medio. De este modo, las tres cuestiones planteadas quedarán respondidas.

Optimización del error medio respecto del umbral

En el desarrollo de la transformada de Haar explicado estamos realizando experimentos en donde para cada valor umbral ε obtenemos un error medio. Estas dos medidas están relacionadas de manera directa. Obviamente nos gustaría tener un error muy bajo descartando muchos coeficientes pero esto es imposible por eso vamos a definir una optimización basada en la búsqueda de un compromiso entre ambas medidas.

Hay otras medidas que están involucradas y guardan una relación con éstas, como el porcentaje de coeficientes descartados y el factor de compresión. Muchas veces podemos aumentar el porcentaje de descarte aumentando también el factor de compresión (que es el objetivo perseguido) sin aumentar apenas el error medio. De este modo es necesario hacer una buena elección del valor umbral del algoritmo **thresholding** que determinará el porcentaje de descarte ya que la ganancia obtenida es en todos los sentidos. De esta forma se pretende encontrar un equilibrio entre tener un buen factor de compresión y tener una imagen recuperada muy similar a la primitiva.

Para tomar una elección inteligente se ha examinado cómo se agrupan los datos en diferentes gráficas. Después de este análisis el punto óptimo es el punto *knee* de la curva que relaciona el porcentaje de descartes y el error. A continuación definimos este concepto y explicamos por qué es la solución tomada.

Un punto *knee* de una curva es aquel en el que la tangente a la curva forma un ángulo de 45° con el eje de abscisas. Otra interpretación del punto *knee* de una curva es que es aquel donde la curvatura tiene un máximo local, si existe tal punto. En otras palabras, la curva tiene una curvatura más baja a cada lado de dicho punto.

Algorithm 6 Calcula el umbral óptimo para la transformada de Haar

```

procedure OPTIMIZATION(function)
    threshold  $\leftarrow$  Lista de umbrales adecuados al ejemplo
    errs  $\leftarrow$  [ ], pers  $\leftarrow$  [ ]
    for i  $\leftarrow$  0 to len(threshold) do
        err, per  $\leftarrow$  experiment(function, threshold)
        errs.insert(err), pers.insert(per)
    knee  $\leftarrow$  KneeLocator(pers, errs)
    for i  $\leftarrow$  0 to len(threshold) do
        if pers[i] == knee then
            opt_threshold  $\leftarrow$  threshold[i]
    return knee, opt_threshold
```

Éste es el procedimiento diseñado para la búsqueda del óptimo. Así se obtiene un

conjunto de datos bueno para la consecución del punto *knee*. Para hallarlo existen códigos que funcionan muy bien, en este caso se ha usado el módulo `KneeLocator` del paquete `kneed` disponible en Python.

La interpretación que se ha dado del *knee* es el motivo por el que lo elegimos. La idea que se pretende es poder elegir el umbral más alto, o dicho de otras palabras, descartar el mayor número de coeficientes sin que el error empiece a aumentar rápidamente. Y este fenómeno ocurre de manera explícita en la relación entre el porcentaje de descartados y el error medio.

Vamos a mostrar la gráfica de la relación porcentaje de descartados y error. En ella se ha señalado el *knee*.

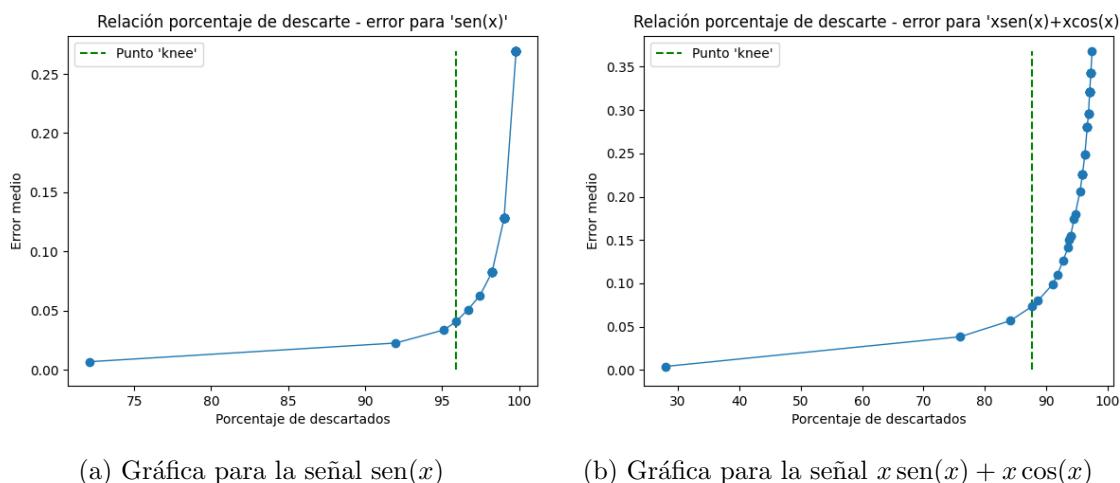


Imagen 3.6: Relación porcentaje de descarte - error

En la del seno, la Imagen 3.6a, el punto *knee* exacto es 95,9 y el umbral que desencadenó ese porcentaje de descartados es 0,016. El error medio bajo estos parámetros es 0,04, muy bajo con un alto coeficiente de descartados que es nuestra mayor premisa.

En la Imagen 3.6b el *knee* es 87,7 y el umbral asociado también es 0,016. El error medio usando estos parámetros es 0,07 que aunque es bajo es considerablemente mayor que en la anterior.

Aunque el error de la del seno era peor que el de la otra función realmente era porque en la segunda se descartan menos coeficientes y por eso en la optimización al buscar un equilibrio ha resultado mejor la del seno.

Vamos a mostrar en último lugar las gráficas del error en función del umbral para las dos funciones f y g . En ellas se señalan los umbrales asociados al punto *knee* (que es un porcentaje de descartados).

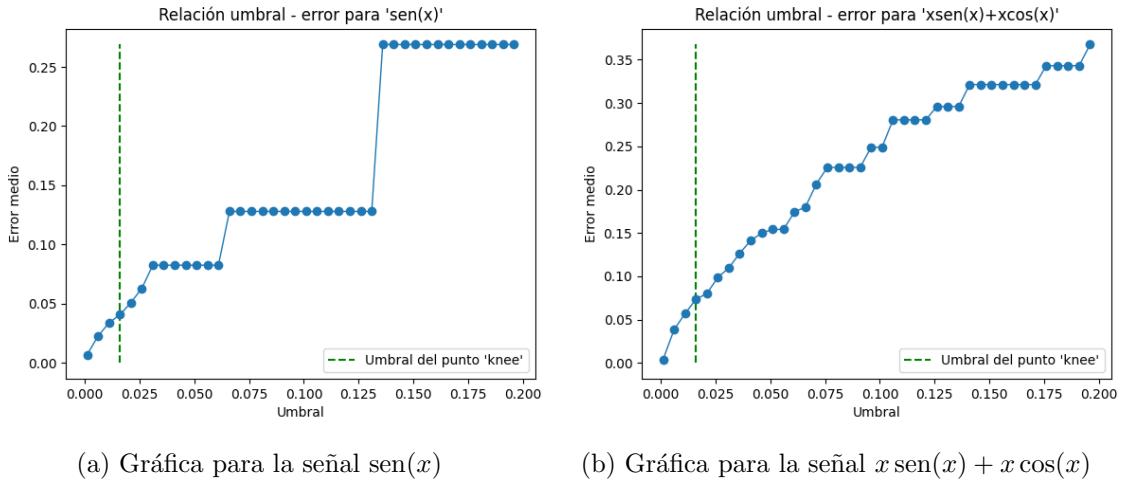


Imagen 3.7: Relación umbral - error

Llama a la vista que ya no se obtiene una función parabólica. Muchas veces un aumento pequeño del umbral no descarta más coeficientes y por eso encontramos una gráfica escalonada. En la Imagen 3.7 este fenómeno ocurre con menor frecuencia. Seguramente los coeficientes son más variados y los cambios de umbral, a pesar de ser pequeños, producen un aumento de descartados agrupando los datos de una manera más lineal.

Optimización del error medio respecto de N

Como ya se anunció otra optimización es bajo el parámetro N o número de trozos en los que se discretiza la señal. Los experimentos que se van a realizar son bajo un umbral fijo y razonable. En este caso se ha usado $\varepsilon = 0,1$ para las dos señales. Es conveniente un umbral alto que descarte bastantes coeficientes para darle peso al efecto de N .

En este caso en las pruebas que se han realizado y analizado se obtienen gráficas convexas decrecientes. Es lógico que sea así. Un número de trozos pequeño conlleva un error grande y un valor de N elevado un error pequeño. Sin embargo el intervalo de valores de N sobre el que ejecutar el experimento debe tener un N inicial aceptable en el sentido de que si es muy pequeño la discretización de la señal es tan mala que no tendría sentido tener un candidato así. El intervalo escogido para $N = 2^n$ es con n en $\{6, \dots, 16\}$ luego el N inicial es 64 trozos.

Como los valores de N van a ser muy grandes seguramente resulte muy útil usar escala logarítmica en el eje de abscisas. Es decir, mostrar n en vez de N para mejorar la visualización y el cálculo del *knee*.

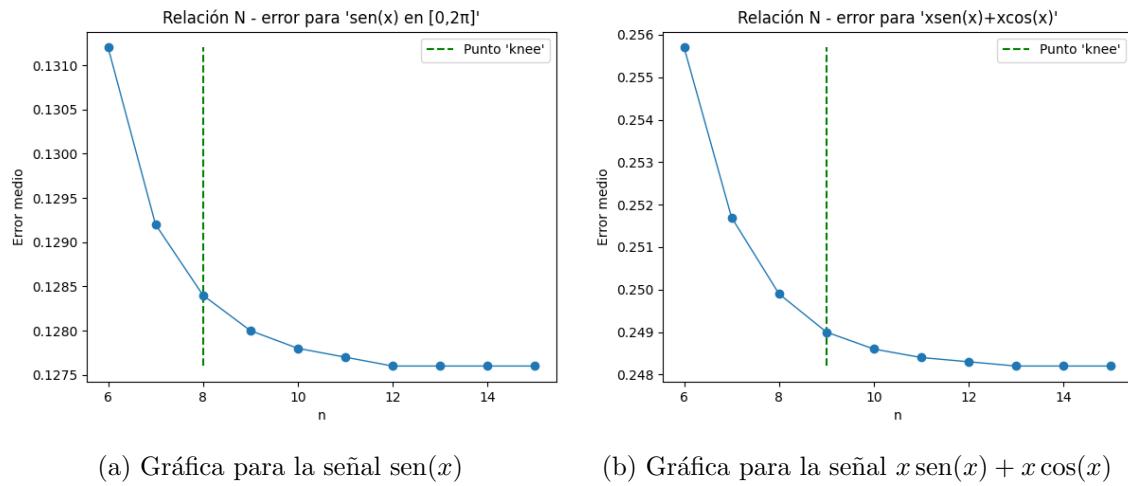


Imagen 3.8: Relación N - error

La primera observación son los óptimos. Para la Imagen 3.8a el óptimo es $n = 8$ ($N = 256$) y el error medio 0,13 y en la Imagen 3.8b el óptimo es $n = 9$ ($N = 512$) y el error medio 0,25.

Una apreciación relevante entre las dos optimizaciones realizadas es que la del umbral reduce mucho más el error que la de N . Podemos notar esto en el eje de ordenadas, las escalas son diferentes. En esta optimización se mueve sobre la misma décima por lo que la optimización del umbral es más efectiva.

Optimización del error medio respecto de p

La última optimización es bajo el parámetro p de normalización. El gran trabajo realizado al principio del desarrollo informático permite que esto sea discutido fácilmente. Se dispone del diseño e implementación de un procedimiento que proporciona de manera eficiente los coeficientes normalizados en cualquier $L_p[0, 1]$.

Se elige un conjunto de valores de p sobre el que ejecutar el experimento. Éste conjunto es $[0, 2, 5, 10, 20, 40]$. Recordemos que el valor $p = 0$ computacionalmente quiere decir que los coeficientes están sin normalizar. Obviamente para hacer un análisis objetivo es necesario que el porcentaje de descarte para cada p sea el mismo.

Si fijo un umbral entonces en cada normalización se quita una cantidad de coeficientes diferente. Es más podemos decir que a mayor p menor valor tienen los coeficientes y mayor es la cantidad que atrapa el umbral. Sin duda, es momento de usar el algoritmo de m -términos `m_term`. Vamos a elegir un m que descarte el 80% de los coeficientes. De esta manera ponemos a todos los L_p en las mismas condiciones.

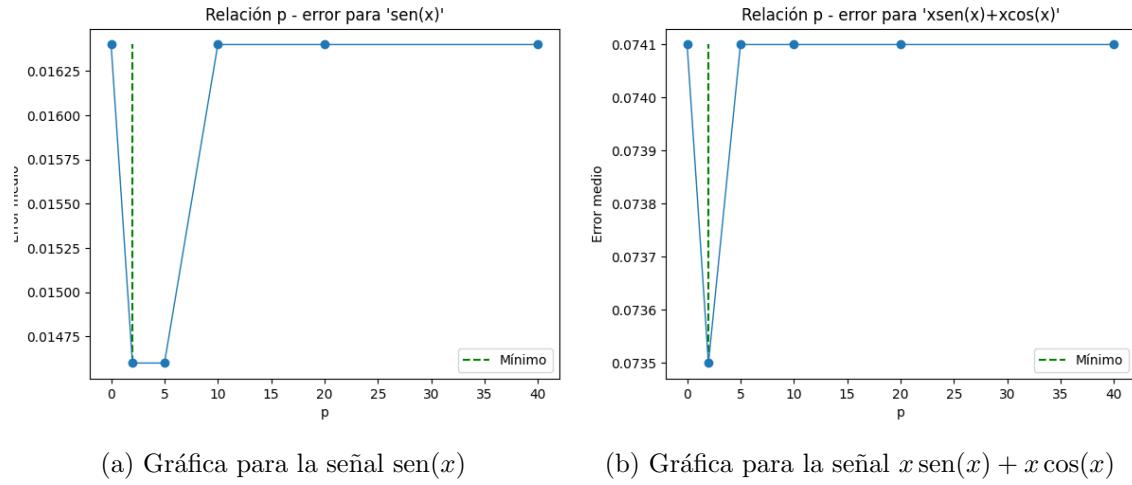


Imagen 3.9: Relación p - error

En la imagen se ha señalado el mínimo pues es el óptimo que se va a considerar. Es obvio que el peor de los casos es $p = 0$, coeficientes sin normalizar. En este caso los coeficientes no reflejan su importancia con respecto al resto y `m_term` los trata por igual.

La mejor elección es la que se ha llevado a cabo en el desarrollo del trabajo, $p = 2$. A partir de ese p el error empieza a aumentar. En conclusión la mejor elección para estas señales es $p = 2$.

3.3. Transformada de Haar para compresión de imágenes

El segundo ejemplo de compresión de información que vamos a llevar a cabo es para imágenes. Esta sección es la parte central del desarrollo informático y pone de manifiesto la potencia de toda la teoría matemática desarrollada mediante un ejemplo tan visual [12, 13].

Vamos a seguir un esquema de trabajo análogo al de las señales. En primer lugar realizaremos un diseño de la solución explicando mediante ejemplos sencillos cómo se adapta la transformada de Haar discreta a este tipo de compresión. A continuación se comentan los aspectos más fundamentales de la implementación y se analizan los resultados. En última instancia se optimizan parámetros del problema para mejorar el error y la compresión.

El código de esta sección está en la carpeta `img_code` a la que se puede acceder a través de https://github.com/Carlossamu7/TFG/tree/master/img_code. A parte del código en Python también están disponibles también los conjuntos de imágenes usados.

Tal y como se hizo en el ejemplo de las señales se van a plantear una serie de cuestiones

que surgen al abordar esta aplicación. A lo largo del desarrollo nos gustaría responder a dichas preguntas.

- (I1) ¿Cómo es la compresión para imágenes en blanco y negro? ¿Y en color?
- (I2) ¿La matriz de los coeficientes respecto de la base de Haar es sólo números o si la pintamos observamos algo con un cierto sentido geométrico?
- (I3) ¿La imagen obtenida de la diferencia entre la imagen original y la revertida después de la compresión es una medida del error?
- (I4) ¿Se puede optimizar la relación umbral-error?
- (I5) ¿Cuál es la mejor elección de espacio de normalización $L_p[0, 1]$?

Como referencia en el estado del arte de la compresión de imágenes tenemos el estándar JPEG. A lo largo de la sección se dará una pequeña comparativa con él.

3.3.1. Diseño de la solución

Las imágenes que vamos a tratar no van a tener formato o compresión alguna y las veremos como una matriz de píxeles. Para ello vamos a extender el procedimiento explicado para listas a arrays 2-dimensionales o matrices.

En la transformada de Haar se hallan promedios y diferencias. En el caso de imágenes denominamos a dichas diferencias como *coeficientes de detalle*, y de ahora en adelante usaremos los términos *diferencia* y *detalle* indistintamente. Podemos considerar los promedios como una representación de resolución aproximada de la imagen original, y los detalles como los datos necesarios para reconstruir la imagen original a partir de esta resolución aproximada.

Si los píxeles de la imagen están correlacionados, la representación aproximada se parecerá a los píxeles originales, mientras que los detalles serán pequeños. Este es el motivo por el cual la compresión de imágenes con la transformada de Haar usa promedios y detalles. Además el éxito de la compresión se debe a la similitud entre píxeles adyacentes, lo cual hace que haya bastantes ceros en las diferencias aumentando el número de coeficientes descartados.

Procedimiento para una matriz

Cuando tratamos con un array 2-dimensional nos encontramos en un caso más complejo. Hay dos vías de llevarlo a cabo:

- Hacer una iteración sobre las filas tal y como lo hemos explicado y luego sobre las columnas y así sucesivamente.

- Hacer todas las iteraciones sobre las filas y una vez hemos acabado hacer las iteraciones sobre las columnas.

Aquí vamos implementar la segunda vía pues una vez que tengamos implementado el procedimiento por filas para hacerlo por columnas sólo hemos de transponer la matriz y volverlo a aplicar. Claramente de la segunda forma realizamos muchas menos transposiciones y por tanto resulta más acertada.

Para una matriz de tamaño 4×4 el desarrollo sería como sigue:

$$\begin{pmatrix} 12 & 12 & 8 & 8 \\ 12 & 12 & 8 & 8 \\ 10 & 10 & 8 & 8 \\ 10 & 10 & 8 & 8 \end{pmatrix} \rightarrow \begin{pmatrix} 10 & 2 & 0 & 0 \\ 10 & 2 & 0 & 0 \\ 9 & 1 & 0 & 0 \\ 9 & 1 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 9,5 & 1,5 & 0 & 0 \\ 0,5 & 0,5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (3.4)$$

donde la primera matriz es la original, la segunda la resultante de aplicar el procedimiento por filas y la última la obtenida de aplicar el proceso por columnas.

En último lugar, un ejemplo de una matriz 8×8 que presenta más diferencias en los números. Sea la matriz de píxeles:

$$\begin{pmatrix} 12 & 12 & 12 & 12 & 8 & 8 & 10 & 10 \\ 12 & 12 & 12 & 12 & 8 & 8 & 10 & 10 \\ 10 & 10 & 10 & 10 & 8 & 8 & 10 & 10 \\ 10 & 10 & 10 & 10 & 8 & 8 & 10 & 10 \\ 22 & 22 & 22 & 22 & 8 & 8 & 16 & 16 \\ 22 & 22 & 22 & 22 & 8 & 8 & 16 & 16 \\ 22 & 20 & 20 & 20 & 14 & 14 & 4 & 4 \\ 22 & 20 & 20 & 20 & 14 & 14 & 4 & 4 \end{pmatrix},$$

entonces sus coeficientes en la base de Haar son:

$$\begin{pmatrix} 12,9375 & 3,1875 & 0,125 & -0,25 & 0,25 & 0 & 0 & 0 \\ -2,9375 & -2,1875 & -0,125 & -0,75 & -0,25 & 0 & 0 & 0 \\ 0,5 & 0,5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1,125 & -0,375 & -0,25 & -4,5 & -0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.5)$$

Nuestro objetivo en esta sección será escoger de diferentes formas los mejores coeficientes. Algunas opciones son:

- Limitarnos simplemente a escoger el número m de coeficientes que queremos quedarnos y hacer una aproximación de m -términos como hemos visto en el desarrollo matemático del trabajo.
- Escoger un valor umbral $\varepsilon > 0$ que sea pequeño y poner ceros en aquellos valores de la matriz o píxeles cuyo valor absoluto sea más pequeño que dicho umbral.

Para ello, como ya sabemos, usaremos los coeficientes respecto del sistema de Haar normalizado. Como nuestro caso es en dos dimensiones la norma de la función de Haar en el píxel (i, j) es el producto $\|h_i\| \|h_j\|$.

La matriz de los coeficientes respecto del sistema de Haar normalizado en $L_2[0, 1]$ de 3.4 es idéntica a ella porque los únicos píxeles distintos de 0 tienen norma uno. Para el ejemplo 3.5 (redondeando a 4 decimales) realizamos la normalización:

$$\begin{pmatrix} 12,9375 & 3,1875 & 0,0884 & -0,1768 & 0,125 & 0 & 0 & 0 \\ -2,9375 & -2,1875 & -0,0884 & -0,5303 & -0,125 & 0 & 0 & 0 \\ 0,3536 & 0,3536 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,7955 & -0,2652 & -0,125 & -2,25 & -0,1768 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Estos nuevos coeficientes son los que vamos a usar. Usando los algoritmos unidimensionales normalizados la extensión a dos dimensiones proporciona los coeficientes normalizados.

Antes de seguir vamos a analizar la complejidad de calcular los coeficientes de una matriz respecto de la base de Haar normalizada de esta forma en función del tamaño de la matriz. Si la matriz tiene $N = 2^n$ filas y $M = 2^m$ columnas entonces:

$$N \cdot (5M - 4) + M \cdot (5N - 4) = 10NM - 4N - 4M.$$

Si $N = M$ entonces la complejidad es $10N^2 - 8N$, orden cuadrático. En el ejemplo 3.4 el número de operaciones aritméticas es 128 y el del ejemplo 3.5 es 576.

3.3.2. Algoritmos

Hemos visto cómo se calcula la transformada de Haar para obtener los coeficientes respecto de la base. No es un procedimiento excesivamente complejo. Vamos a exponer las explicaciones más importantes y mostrar el pseudo-código de los métodos principales.

En primer lugar comenzamos con la obtención de los coeficientes, a continuación se analizará el proceso para revertir la imagen original y por último la aproximación greedy. Una vez estén todas las extensiones de los algoritmos implementadas el experimento es análogo al del Algoritmo 5.

Obtención de los coeficientes de Haar de una imagen

Tal y como anticipábamos esto se puede hacer de varias maneras. Los dos enfoques se denominan *descomposición estándar* y *descomposición piramidal*. Nosotros nos vamos a centrar en entender y examinar el primero de ellos.

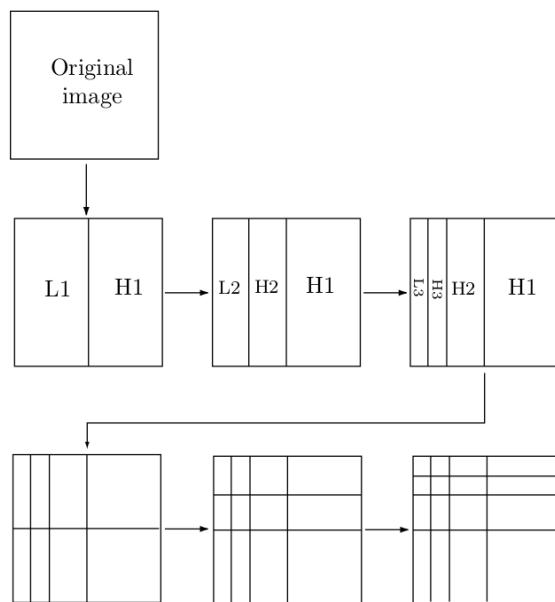


Imagen 3.10: Procedimiento de obtención de los coeficientes de la base de Haar

La Imagen 3.10 da una idea visual de lo que pretendemos hacer. Comenzamos calculando la transformada de cada fila de la imagen. Esto da como resultado una imagen transformada donde la primera columna contiene promedios y todas las otras columnas contienen diferencias. A continuación se calcula la transformación de cada columna. Esto da como resultado un valor promedio en la esquina superior izquierda, con el resto de la fila superior que contiene promedios de diferencias, y con todos los demás valores de píxeles transformados en diferencias.

Las transformaciones wavelet, como la transformación de Haar, son *transformaciones de subbanda*. Dividen la imagen en regiones de modo que una región contenga números grandes (promedios en el caso de la transformación de Haar) y las otras regiones contengan números pequeños (diferencias). Sin embargo, estas regiones denominadas subbandas, son más que simples conjuntos de números grandes y pequeños ya que reflejan las diferen-

cias geométricas presentes en la imagen.

En una transformada wavelet discreta, la mayoría de los coeficientes wavelet son detalles (o diferencias). Los detalles en los niveles inferiores representan los detalles finos de la imagen. A medida que avanzamos en el nivel de subbanda, encontramos detalles que corresponden a características de imagen más gruesas.

El algoritmo que vamos a implementar es para el cálculo de coeficientes de Haar de una imagen de tamaño $2^p \times 2^q$ para cualesquiera p y q naturales. El único parámetro que tiene es la imagen original y ésta pueda ser en escala de grises o en color.

Algorithm 7 Coeficientes de Haar de una imagen

```

procedure HAAR_IMAGE(img, p)                                ▷ Imagen en B/N
    if es_BN(img) then
        by_row  $\leftarrow$  haar_row(img, p).
        haar_img  $\leftarrow$  haar_row(by_rowT, p).                ▷ (T) es traspuesta
        haar_img  $\leftarrow$  haar_imgT.
    else                                                 ▷ Imagen en color
        for k  $\leftarrow$  0 to 2 do
            haar_img[k]  $\leftarrow$  haar_image(img[k], p).
    return haar_img
```

La función `es_BN` devuelve un booleano que dice si la imagen es en blanco y negro o en color. Esta parte es sencilla e intuitiva pues teniendo la función `haar_row` que calcula los coeficientes por filas de una matriz tenemos el trabajo hecho.

Algorithm 8 Coeficientes de Haar de una imagen por filas

```

procedure HAAR_ROW(img, p)
    for i  $\leftarrow$  0 to len(img) do
        row[i]  $\leftarrow$  haar_transform(img[i], p)
    return row
```

Esta parte tampoco es complicada, donde reside el interés es en el cálculo de los coeficientes para una lista, que puede ser una fila o una columna. La función `haar_transform` ya la tenemos, es el Algoritmo 1.

Estos procedimientos son válidos para imágenes con tamaños $2^p \times 2^q$ con p y q naturales. Si la imagen no posee un tamaño así también podemos aplicar la compresión extendiendo a la de tamaño $2^p \times 2^q$ más pequeña dentro de la cual esté contenida.

La extensión se puede realizar de dos maneras:

- Rellenando con ceros los nuevos píxeles.
- Rellenando con el último valor de la fila/columna los nuevos píxeles de esa fila/columna.

Se ha elegido la primera opción, extensión por ceros. Aunque cabe destacar que con la segunda las diferencias se anularían.

Restauración de una imagen a partir de sus coeficientes de Haar

Comenzamos con el procedimiento para una matriz (puede ser en escala de grises o en color) en donde se calcula la inversa de la transformada por filas y columnas. Si está en color la función es recursiva y aplica el proceso a los tres canales de la imagen.

Algorithm 9 Restaura una imagen a partir de sus coeficientes de Haar

```

procedure REVERSE_IMAGE(haar_img, p)
    if es_BN(haar_img) then                                ▷ Imagen en B/N
        by_row  $\leftarrow$  reverse_row(haar_img, p).
        rev_img  $\leftarrow$  reverse_row(by_rowT, p).
        rev_img  $\leftarrow$  rev_imgT.
    else                                                 ▷ Imagen en color
        for k  $\leftarrow$  0 to 2 do
            rev_img[k]  $\leftarrow$  reverse_image(haar_img[k], p).
    return rev_img

```

De nuevo, teniendo un procedimiento **reverse_row** que realice el cálculo inverso sobre todas las filas de una matriz, transponemos adecuadamente y podemos extenderlo para que también se realice sobre las columnas.

Algorithm 10 Restaura una imagen por filas a partir de sus coeficientes de Haar

```

procedure REVERSE_ROW(haar_img, p)
    for i  $\leftarrow$  0 to len(haar_img) do
        row[i]  $\leftarrow$  reverse_haar(haar_img[i], p)
    return row

```

Como observamos el procedimiento es llamar para cada fila de la matriz a la función **reverse_haar** que calcula los valores originales de una lista dados sus coeficientes de Haar. Se recuerda que dicha función es el Algoritmo 3. Se ha usado la notación `[start:stop]` para indicar la parte de la lista que nos interesa, en caso de faltar **start** será cero y si falta **stop** será hasta el final de la lista.

En el caso de que la imagen no tuviese un tamaño $2^p \times 2^q$ y fuese extendida, ahora debemos recortar la parte sobrante y devolverle su tamaño original.

Aproximación greedy

La transformación de Haar es la transformación wavelet más simple, pero incluso este método simple ilustra el poder de la transformación wavelet. Resulta que los bajos niveles de la transformada wavelet discreta contienen las características de la imagen sin importancia, por lo que cuantificar o descartar estos coeficientes puede conducir a una compresión con pérdida que es eficiente y de alta calidad. A menudo, la imagen se puede reconstruir a partir de muy pocos coeficientes de transformación sin ninguna pérdida notable de calidad. El principio fundamental de las imágenes que hace de éste un gran método de compresión es que sea cual sea el píxel que elijamos es muy probable que sus vecinos sean similares a él.

La compresión de imágenes usando la transformada de Haar con pérdida implica el descarte de coeficientes, por lo que definimos el concepto de *ratio de dispersión* para medir la cantidad de coeficientes descartados. La dispersión es el número de coeficientes distintos de cero dividido por el número de coeficientes restantes después de hacer el descarte. Notemos que este número es por tanto mayor o igual que 1 siempre. Cuanto mayor sea el ratio de dispersión, menos coeficientes quedan. Ratios de dispersión más altos conducen a una mejor compresión, pero pueden dar lugar a imágenes mal reconstruidas. Así existe una relación entre los conceptos de ratio de dispersión y factor de compresión.

Los algoritmos de descarte de los coeficientes son los explicados: `thresholding` y `m_term`. Han sido extendidos para su uso en matrices.

3.3.3. Resultados y análisis

Se han seleccionado dos imágenes en escala de grises y dos imágenes en color para testear los resultados de la implementación. Mediante ellas se va a analizar el comportamiento de la transformada de Haar así como las aportaciones que hace a la compresión de información. Este apartado está por tanto ligado a la pregunta (I1).

Imágenes en escala de grises

Vamos a examinar dos casos: el primero para un umbral bajo $\epsilon = 0,005$ y el segundo para un umbral alto $\epsilon = 0,05$. La primera imagen es famosa en el mundo de la Visión por Computador por su historia. Es conocida popularmente como “Lena” y tiene tamaño 512×512 . Empezamos aplicando `thresholding` con $\epsilon = 0,005$.



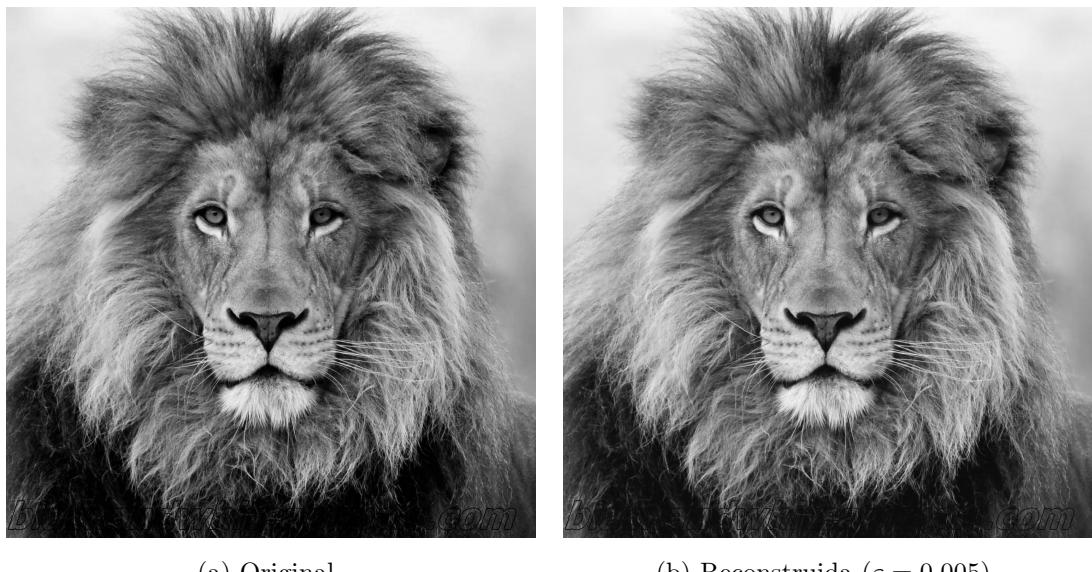
(a) Original

(b) Reconstruida ($\varepsilon = 0,005$)Imagen 3.11: Lena en escala de grises después de `thresholding` con $\varepsilon = 0,005$

El número de píxeles descartados es 122.036, es decir, un 46,55 % de los coeficientes son cero. Sin embargo, a pesar de descartar la mitad de la imagen la diferencia entre la original y la reconstruida es mínima, casi imperceptible al ojo humano.

El error medio por píxel obtenido con estos parámetros es 0,7693 y el ratio de dispersión 1,6595. En una imagen en la que los valores están en el intervalo [0, 255] un error medio inferior a 1 es muy aceptable. Digamos que hemos quitado ruido de la imagen.

Probemos un nuevo experimento con otra imagen de tamaño mayor, 2048×2048 .



(a) Original

(b) Reconstruida ($\varepsilon = 0,005$)Imagen 3.12: León en escala de grises después de `thresholding` con $\varepsilon = 0,005$

De nuevo a simple vista las imágenes son idénticas. El número de píxeles anulados en esta imagen es 3.618.312 que supone un 86,27 %. Haciendo el cociente de la cantidad de coeficientes distintos de cero antes del algoritmo de descarte con respecto a los de después, hallamos el ratio de dispersión, 5,9176.

El error medio para la imagen del león es de 2,191 por píxel, lo cual es bajo teniendo en cuenta que el porcentaje de coeficientes descartados es mayor al que obtuvimos antes. Observamos un alisado de algunas zonas de la imagen sin perder apenas detalles.

En estos tests no existe diferencia entre las imágenes. Es interesante llevar estos algoritmos más el límite y ver cómo sería la reconstrucción llevando un porcentaje muy alto de píxeles a cero.

Elegimos un nuevo umbral $\varepsilon = 0,05$ para la imagen de Lena.



Imagen 3.13: Lena en escala de grises después de **thresholding** con $\varepsilon = 0,05$

El número de píxeles anulados es 246.482, un 94,03 %. El ratio de dispersión de esta imagen es 14,8452 y el error medio entre la original y su aproximación es 4,4168. Aunque es notablemente mayor que el de $\varepsilon = 0,005$, es un error aceptable. El tamaño del archivo original es 1.048.576 bytes y el de la compresión 4.096 bytes por lo que el factor de compresión es 256. Se está consiguiendo comprimir mucho.

A pesar de no tener un error medio demasiado grande, la reconstrucción de la imagen aparece mínimamente pixelada. Que las dimensiones de la original sean pequeñas puede influir en el pixelado.

Volvemos a la imagen del león con $\varepsilon = 0,05$:

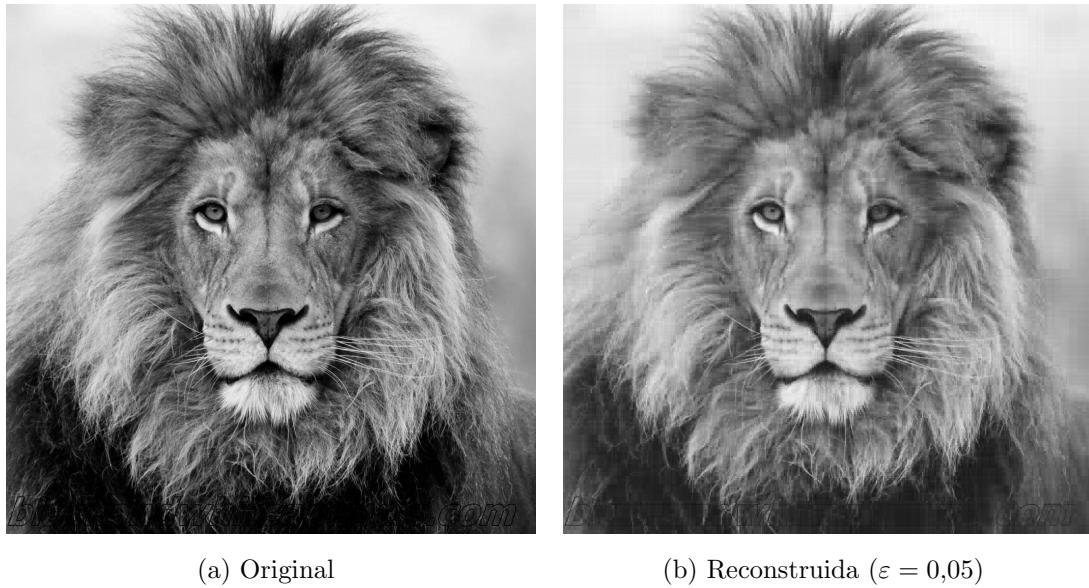


Imagen 3.14: León en escala de grises después de `thresholding` con $\varepsilon = 0,05$

Tenemos un 99,36 % de los coeficientes anulados y un ratio de dispersión muy alto, 127,558. Sin embargo, a pesar de manejar sólo un 1 % de los coeficientes el error de la imagen reconstruida es 8,3062. El factor de compresión es 1024, excelente.

Para este umbral la aproximación del león es mucho mejor. A pesar de tener el doble de error no da sensación se pixelado como la de Lena. En ésta hay una pérdida de oscuridad en el pelo así como un alisado del difuminado grisáceo que rodea el león. La reconstrucción del león es de más calidad que la de Lena. Como finalización de los resultados para imágenes en escala de grises recogemos los diferentes resultados en la siguiente tabla:

Tabla 3.2: Recopilación resultados de imágenes en B/N

Ejemplo	Umbral	Porcentaje de descarte (%)	Ratio de dispersión	Error medio	Factor de compresión
Lena	0,005	46,55	1,6595	0,7693	256
León	0,005	86,27	5,9176	2,191	1024
Lena	0,05	94,03	14,8452	4,4168	256
León	0,05	99,36	127,558	8,3062	1024

Observamos que con un umbral bajo estamos quitando alrededor de la mitad de los coeficientes. Esto se debe a la similitud entre píxeles adyacentes en la imagen original. Con este tipo de umbral el ratio de dispersión también es bajo. Llama la atención que un mismo umbral descarta porcentajes de coeficientes tan distintos en las dos imágenes. Una

posible explicación es la diferencia de tamaño entre ambas. La del león es más grande y en aquellos píxeles alejados de la esquina superior izquierda la norma de la función de Haar es muy pequeña por lo que es razonable que el umbral los atrape con mayor facilidad.

Cuando el umbral es notablemente alto más del 90 % de los coeficientes se descartan, implicando ratios de dispersión muy altos. No obstante el error medio no crece en la misma medida que los parámetros anteriores. Aunque es cierto que ha aumentado, no es un gran error teniendo en cuenta que sólo estamos manejando la información de menos de un 1 % de los píxeles en el caso del león. Esta es la potencia del algoritmo desarrollado.

En general, visualmente las aproximaciones para los casos de B/N han sido muy buenas y las imágenes reconstruidas se asemejan a las originales. Además la compresión es un éxito, los factores obtenidos son excelentes. El factor de compresión de la imagen del león es cuatro veces mayor que el de lena pero es que la imagen del león tiene un número de píxeles 16 veces mayor que la de lena. Que se obtenga el mismo factor para diferentes umbrales seguramente se deba a la asignación de memoria dinámica que Python realiza.

Imágenes en color

Vamos a realizar pruebas muy parecidas a las anteriores pero con imágenes a color, en donde aplicamos todos los procedimientos explicados a cada banda de la imagen. Comenzaremos con pruebas con umbrales bajos y luego lo aumentaremos a un umbral más alto para poder ver cómo afecta a la compresión y reconstrucción de la imagen.

La primera imagen es la versión en color de Lena y usaremos $\varepsilon = 0,005$.



Imagen 3.15: Lena en color después de thresholding con $\varepsilon = 0,005$

El porcentaje de coeficientes descartados ha sido 54,41 % y el ratio de dispersión 2,0673. La reconstrucción de la imagen es muy buena, se observa un alisado en algunas zonas y pequeños cambios de intensidad con respecto a la original. De hecho, los colores de la Imagen 3.15b están mejor que los de la Imagen 3.15a.

El error medio por píxel es 0,7308, lo que confirma la similitud que veíamos. Este error es ligeramente mejor que en la versión en blanco y negro donde era 0,7693. La compresión también es satisfactoria, el factor hallado es 256.

La última imagen que vamos a usar es una fotografía paisajística de la Alhambra de Granada. Además éste va a ser el primer ejemplo de imagen que no es de tamaño $2^p \times 2^q$ con $p, q \in \mathbb{N}$. El tamaño original de la imagen es 663×2000 y por tanto la extendemos por ceros a tamaño 1024×2048 .

Es necesario guardar las dimensiones de la imagen original para posteriormente cuando se recupere la imagen a través de sus coeficientes poder recortarla al tamaño primitivo y así quedarnos con la parte relevante.



(a) Original



(b) Reconstruida ($\varepsilon = 0,005$)

Imagen 3.16: Alhambra después de `thresholding` con $\varepsilon = 0,005$

El umbral usado $\varepsilon = 0,005$ conlleva un porcentaje de descarte del 72,03 %. En es-

te cálculo sólo se ha considerado la zona de la imagen sin extender. El error medio es aceptable, 1,3201.

El ratio de dispersión es 1,6635, que es más bajo comparado a los obtenidos por ahora. La compresión es bastante buena: el tamaño original es 15.912.000 bytes y el de la compresión 24.576 bytes. De este modo, el factor de compresión es 647,461.

Dejando a un lado los resultados numéricos, que son importantes, ambas imágenes son realmente similares a simple vista.

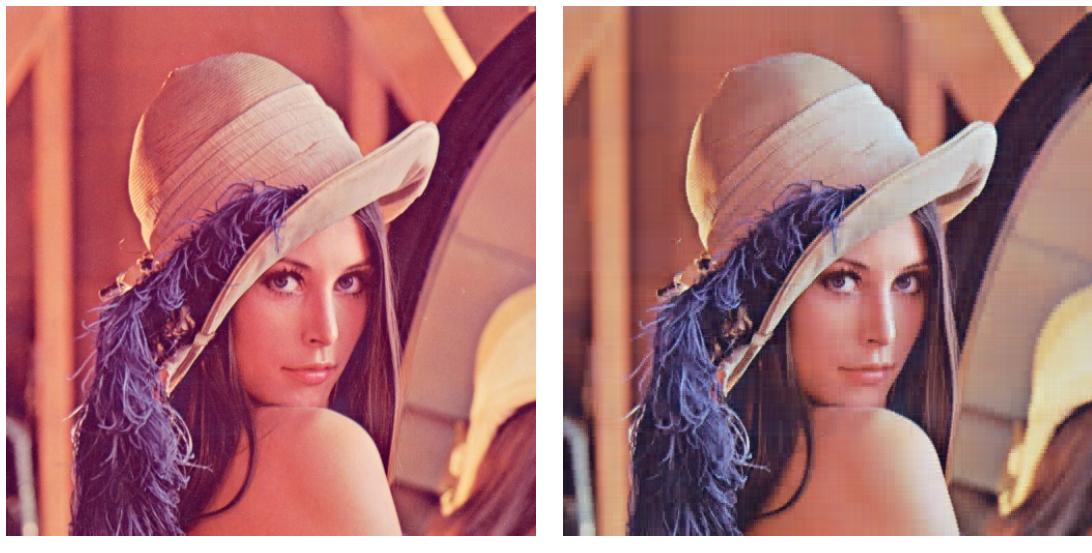
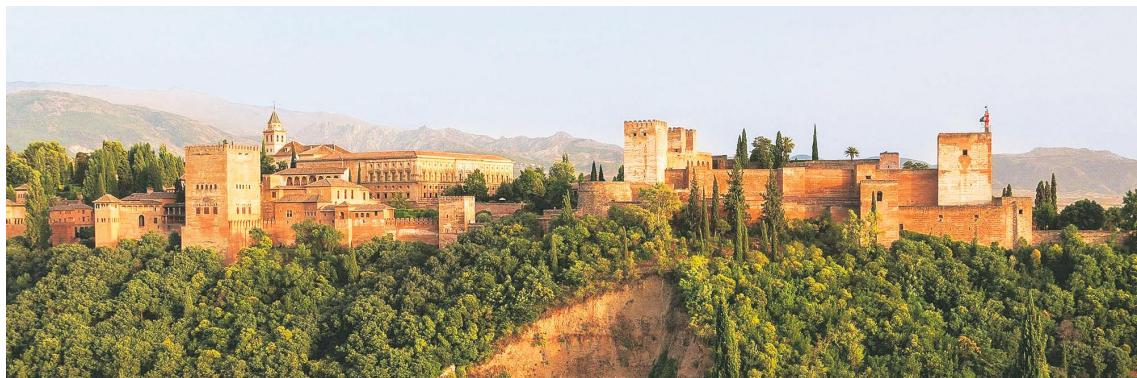


Imagen 3.17: Lena en color después de **thresholding** con $\varepsilon = 0,05$

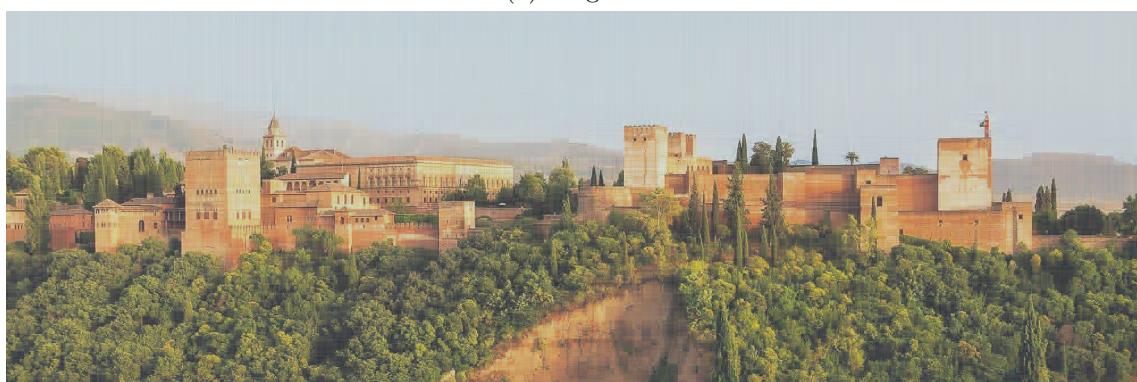
En este ejemplo se han anulado el 94,3 % de los coeficientes provocando un error medio de 4,2148. El ratio de dispersión es 16,5414. El factor de compresión coincide con la versión en gris, 256.

El hecho de que la imagen esté en color afecta la recuperación pues las tonalidades han cambiado bastante entre la imagen original y la recuperada. Un fenómeno que no era perceptible en las versiones en escala de grises. Los indicios que veíamos la Imagen 3.14 de pérdida de oscuridad aquí se ven mucho más aumentados. Esa pérdida de intensidad ahora está ocurriendo en cada canal de la imagen y por eso el efecto es mayor. Aunque el error medio es bajo, este efecto es notable. Esto responde la pregunta (I1).

Por último vamos a aplicar un **thresholding** con un umbral $\varepsilon = 0,05$ al caso de la Alhambra. A través de este ejemplo vamos a comprobar si también ha habido cambios en la tonalidad e intensidad de los colores como ha pasado en el caso de Lena para umbral grande.



(a) Original

(b) Reconstruida ($\varepsilon = 0,05$)Imagen 3.18: Alhambra después de **thresholding** con $\varepsilon = 0,05$

El **thresholding** ha llevado al 96,91 % de los coeficientes a cero. El resultado es un ratio de dispersión de 15,065 y un error medio de 12,198. El error es el más alto que hemos obtenido y es suficiente para hacer visibles ciertos cambios entre las imágenes. Obtenemos un factor de compresión de 647,4609375. Hay que tener en cuenta que se está comprimiendo la imagen extendida (de los coeficientes) la cual es más grande. Sin embargo, al extender por ceros estos se comprimirán muy bien.

Más allá de los números, que son buenos, podemos decir que aunque la reconstrucción de la imagen es buena y se aprecia perfectamente la Alhambra, el color rojizo del amurallado ha disminuido y el cielo se ha atenuado mucho. Asimismo todos los arboles han perdido detalle e intensidad. En general se ha perdido un poco de calidad y profundidad en los detalles de toda la imagen.

En las imágenes en color el efecto más observado es la atenuación de colores y tonalidades. A pesar de realizar un proceso de normalizado después de este proceso al rango [0, 255] no se recupera la misma intensidad que en las imágenes originales.

Para finalizar este subapartado dedicado al análisis de los resultados de imágenes en

color recojo todos los datos de los ejemplos en la siguiente tabla:

Tabla 3.3: Recopilación resultados de imágenes en color

Ejemplo	Umbral	Porcentaje de descarte (%)	Ratio de dispersión	Error medio	Factor de compresión
Lena (color)	0,005	54,41	2,0673	0,7308	256
Alhambra	0,005	72,03	1,6635	1,3201	647,461
Lena (color)	0,05	94,3	16,5414	4,2148	256
Alhambra	0,05	96,91	15,065	12,198	647,461

Los resultados numéricos son similares a los obtenidos en los casos en escala de grises. Tal y como allí ocurría con umbral $\varepsilon = 0,005$ se descarta más de la mitad de la imagen. Con umbrales altos corroboramos que la transformada de Haar funciona estupendamente convirtiéndose en una herramienta de compresión muy potente. Sigue siendo impactante que con menos del 6 % de los píxeles se pueda reconstruir una copia fiel de la imagen.

Todos los errores medios son aceptables. En general, ratios de dispersión altos conducen a factores de compresión altos. Encontramos una dependencia del factor de compresión con el tamaño de la original, más allá de la fórmula de cálculo en la cual están conectados. El factor de compresión de la Alhambra es considerablemente mayor al de Lena pero menor que el del león (blanco y negro) y a las dimensiones originales les ocurre lo mismo. El tamaño en bytes de la compresión es similar en diferentes casos por lo que aquellos cuya imagen original es más grande, obtienen por la forma de cálculo, un mejor factor.

Quiero destacar el caso de la imagen de la Alhambra porque su porcentaje de descartados es similar a otras en las que se usó el mismo umbral y sin embargo el error medio es más alto. Una posible explicación es que es una imagen especial en el sentido de que era rectangular y no tenía tamaño $2^p \times 2^q$ y había que extenderla y aplicar el proceso a la extendida. Eso puede aumentar levemente el error.

Imágenes de los coeficientes respecto de la base de Haar

Una pregunta intrigante a hacerse es (I2), analizar si los coeficientes tienen significado geométrico.

Cuando los coeficientes están sin normalizar sí que se puede observar algo en las imágenes. No es necesario aplicar **thresholding** pero hace una limpieza de la imagen quedándose con detalles más gruesos. Además así podemos estudiar su efecto pues vamos a elegir, como siempre, un umbral pequeño y uno grande (ambos de escala mayor a los usados hasta ahora porque la imagen está sin normalizar). Por tanto, procedemos a mostrar las imáge-

nes de los coeficientes de varias transformadas (sin normalizar), analizar qué ocurre y dar las explicaciones adecuadas.

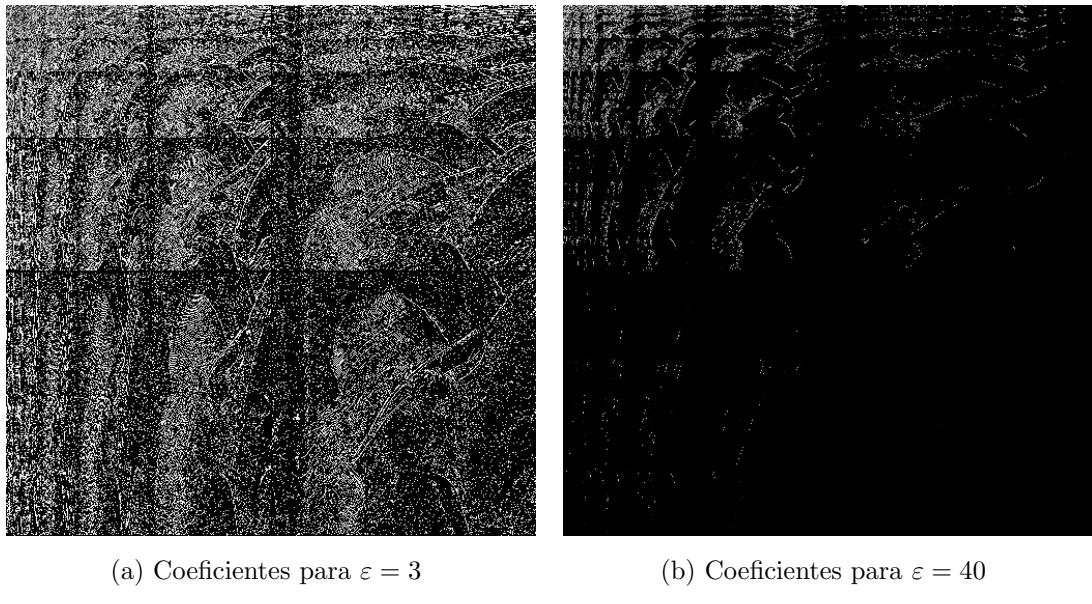


Imagen 3.19: Comparación imagen de coeficientes de Lena para distintos valores de ε

Sorprendentemente observamos la silueta de la imagen. Hay tantas como iteraciones hemos realizado en la transformación de Haar, tanto por filas como por columnas. Dependiendo del valor umbral usado es claro que el nivel de detalle desaparece, especialmente en aquellos píxeles donde la norma es más pequeña.

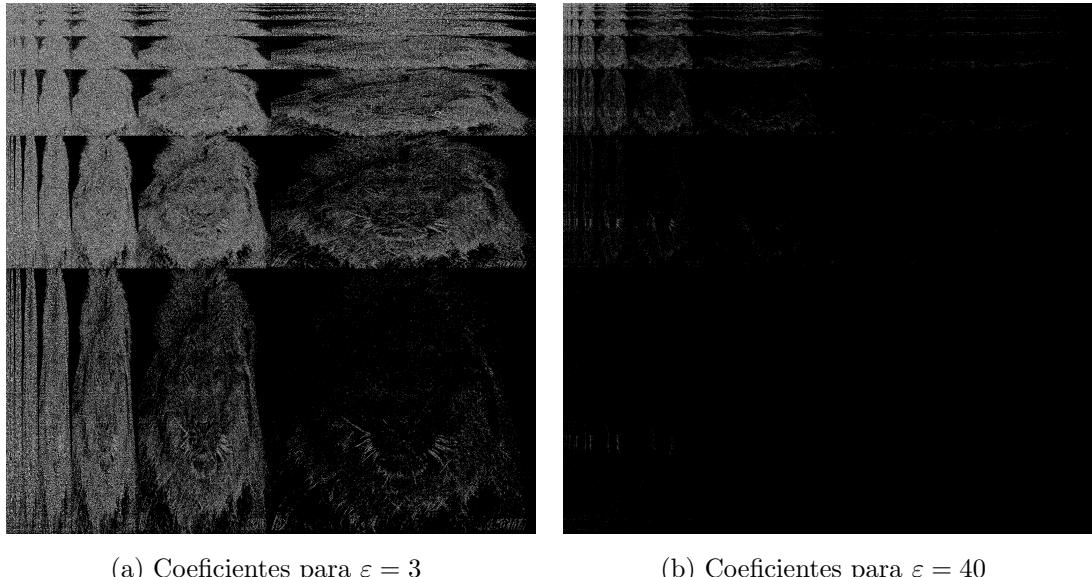


Imagen 3.20: Comparación imagen de coeficientes del león para distintos valores de ε

En la imagen del león ocurre lo mismo. La explicación consiste en que para una función 2D la derivada parcial es

$$\frac{\partial f(x, y)}{\partial x} = \lim_{\varepsilon \rightarrow 0} \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon},$$

pero en el caso de datos discretos podemos aproximarla por

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x + 1, y) - f(x, y)}{1}.$$

Nosotros estamos haciendo la diferencia de los valores de los píxeles (x, y) y $(x+1, y)$, es decir, el opuesto de esa derivada parcial. Lo mismo ocurre por columnas. Aunque dividimos entre 2 no es suficiente para distorsionar el efecto del gradiente. El significado geométrico del gradiente en una imagen es el cambio de intensidad en el color en dicha imagen. Éste es el motivo por el cual observábamos la silueta geométrica de las figuras de las imágenes de los coeficientes que hemos estado tratando.

Cuando los coeficientes son los normalizados, la fórmula del gradiente se ha distorsionado hasta tal punto que no se puede observar este hecho.

En las franjas horizontales tenemos el cambio de intensidad en la dirección x y en las verticales el cambio en y .

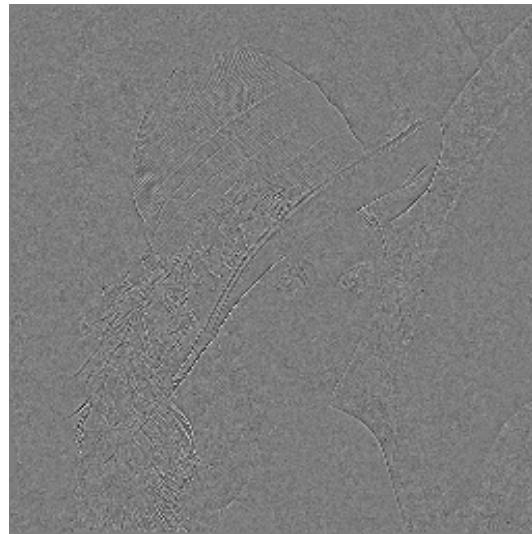


Imagen 3.21: Imagen normalizada de la submatriz inferior derecha de Lena

Imagen diferencia de la original y la revertida

Sea la imagen de tamaño $N \times M$ $A = (a_{ij})_{i=1,j=1}^{N,M}$ y sea $\widehat{A} = (\widehat{a}_{ij})_{i=1,j=1}^{N,M}$ la revertida después de usar alguno de los algoritmos de aproximación sobre la transformada de Haar.

Si construimos $D = A - \hat{A}$ la imagen diferencia de ambas, ¿qué significado tiene?, ¿es una medida del error? Así, en este apartado vamos a responder a la pregunta (I3).

Si construimos $D = (d_{ij})_{i=1,j=1}^{N,M}$ el píxel (i, j) viene dado por $d_{ij} = a_{ij} - \hat{a}_{ij}$. Lógicamente si se usa compresión sin pérdidas (no pasando ningún algoritmo de aproximación sobre la transformada) entonces D va a estar llena de ceros y va a ser una imagen uniformemente negra.

En el momento en el que se aplica un algoritmo de aproximación y la compresión es con pérdida esta diferencia deja de ser cero. La aproximación quita los detalles finos de la imagen conservando los gruesos. Al quitar las frecuencias bajas es probable que en la imagen diferencia queden las frecuencias altas.

Se va a mostrar la imagen diferencia de dos ejemplos: para gris Lena y para color la Alhambra. Se ha usado el umbral más alto $\varepsilon = 0,05$.

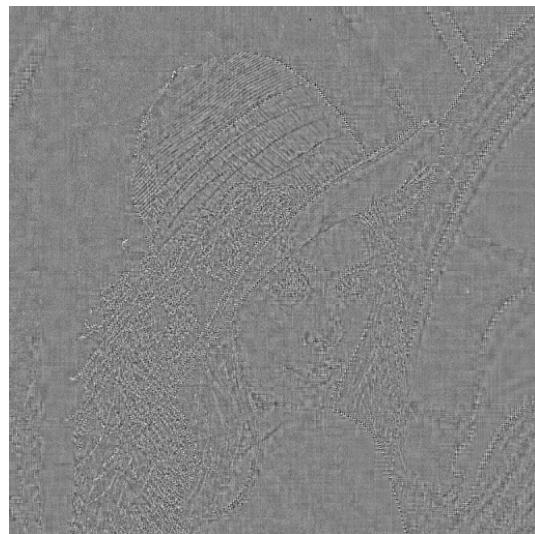


Imagen 3.22: Imagen diferencia para la imagen Lena

A continuación el caso de la Alhambra.

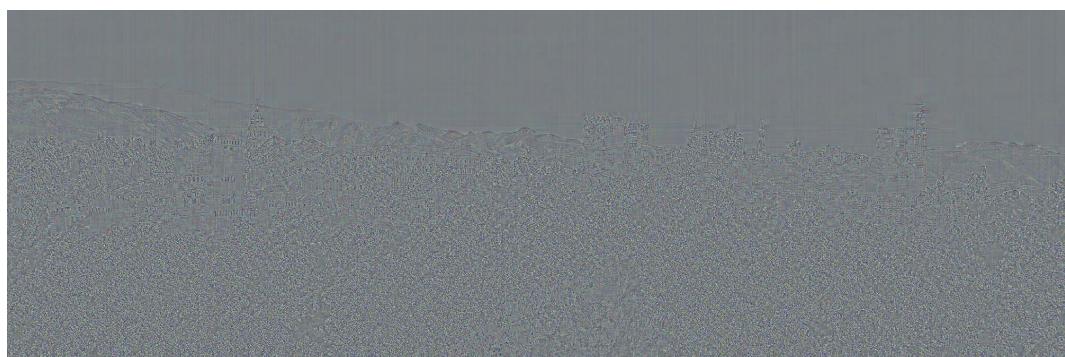


Imagen 3.23: Imagen diferencia para la imagen Alhambra

Efectivamente en ambos casos el efecto es el anunciado. Es lógico que para umbrales mayores se visualice la silueta de una manera más pronunciada y con umbrales pequeños la imagen queda más lisa. Si todos los valores son cero entonces la normalización al rango [0, 255] no haría efecto y la imagen sería negra.

La conclusión de este apartado dedicado a los resultados y al análisis de los mismos es que si el número de coeficientes que descartamos está en torno al 50 % las diferencias entre la imagen original y la recuperada son mínimas y mayoritariamente imperceptibles al ojo humano. Este tipo de aproximación no tiene problema en llevarlo al límite y descartar un porcentaje de coeficientes del 90 % o 95 %. En la imagen recuperada se habrán perdido detalles y calidad pero se identificará bastante bien el contenido. Las imágenes a color son mucho más sensibles y se harán visibles los cambios de tonalidad e intensidad.

Los resultados obtenidos son muy similares a los del algoritmo JPEG (Joint Photographic Experts Group). Dicho algoritmo es complejo y sigue una serie de pasos. Uno de ellos es la transformación discreta de coseno. En el año 2000 la Organización Internacional de Normalización (ISO) modificó el algoritmo cambiando la transformada del coseno por una transformada wavelet y la transformada de Haar es una transformada wavelet.

3.3.4. Optimización del error medio y la compresión

De manera análoga a como se hacía en el caso de las señales se puede optimizar la relación entre el error que se obtiene y la cantidad de información a almacenar. El parámetro que influye directamente en estas dos medidas es el umbral ε , que es el caso de la pregunta (I4). Aquí no existe posibilidad de optimizar N pues ya viene fijo por las dimensiones de la imagen. Sin embargo, sí se puede estudiar cuál es el mejor espacio para la normalización, la elección de p , pregunta (I5). En este apartado estudiaremos las optimizaciones en función del umbral y p .

Optimización del error medio respecto del umbral

Hasta ahora hemos realizado experimentos que consisten en dada una imagen y un umbral realizar la transformada de Haar, elegir los coeficientes mediante aproximación greedy y aplicar la transformada inversa. De este experimento se obtienen diferentes medidas. Las más importantes para la optimización son el error y el porcentaje de descartes.

Los ejemplos que se ha decidido usar para la optimización son la imagen de Lena en versión gris y la de la Alhambra en color, la cual había que extenderla y recortarla en cada ejecución del experimento. Es innegable que la búsqueda del óptimo necesita un tiempo computacional alto pero admisible.

Observamos en primera instancia las gráficas de la relación porcentaje de coeficientes descartados y error medio para Lena. Como era de esperar por los resultados de las señales, el óptimo va a ser el *knee*.

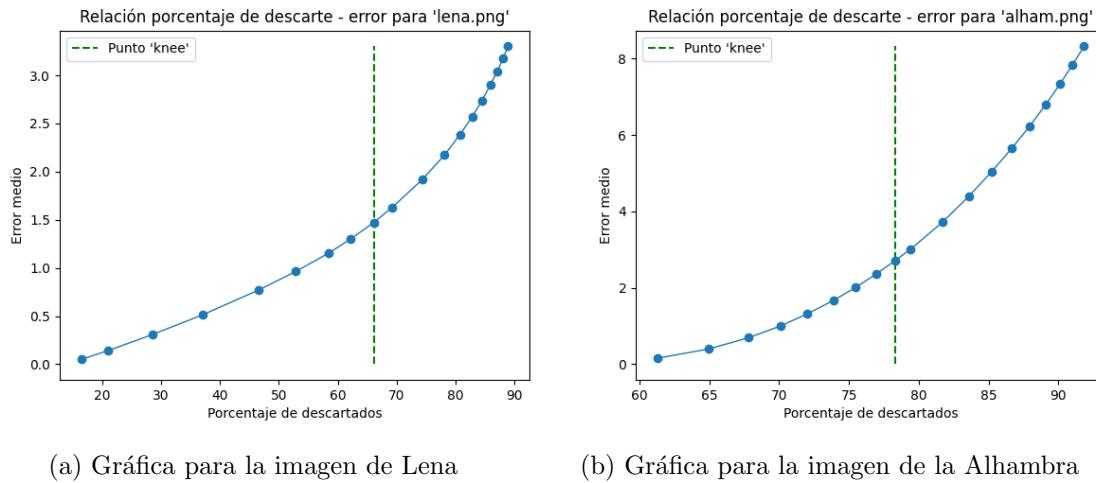


Imagen 3.24: Relación porcentaje de descarte - error

Para la Imagen 3.24a el punto *knee* exacto es 66,14 % y para la 3.24b es 78,28 %. Se han realizado pruebas sobre otras imágenes y también se obtienen porcentajes óptimos parecidos. Aún así, es importante darle el umbral adecuado a cada imagen y tenemos un proceso definido para hallarlo.

Una vez que sabemos el porcentaje de clasificación óptimo (en el sentido definido) sabemos cuál es el umbral que desencadenó ese experimento y ese porcentaje y error. La gráfica de la relación umbral y error se expone a continuación.

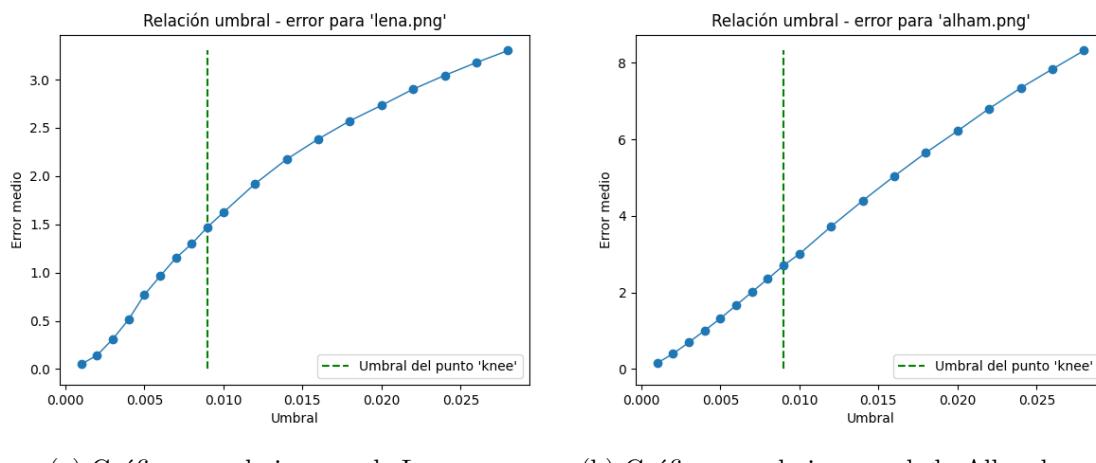


Imagen 3.25: Relación umbral - error

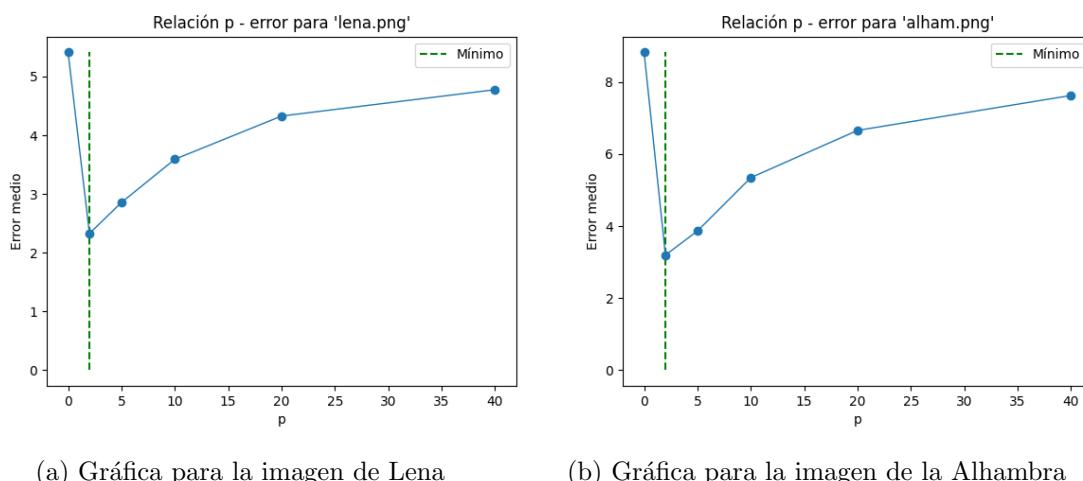
Del mismo modo que se hacía con las anteriores se ha señalado el umbral asociado al punto *knee*. En esta nueva gráfica los datos están agrupados de una manera diferente, más lineal. La gráfica adecuada para la consecución del óptimo era la anterior. El umbral asociado al *knee* de ambas imágenes casualmente coincide: $\varepsilon = 0,009$.

En conclusión, este procedimiento nos determina el punto donde hay un equilibrio entre la compresión y el error. Si preferimos una mayor compresión y aceptamos un mayor error medio podemos seleccionar umbrales mayores al umbral asociado al *knee*. Si por el contrario queremos más calidad en la imagen aunque tengamos que almacenar más coeficientes elegiremos umbrales menores al asociado al *knee*.

Optimización del error medio respecto de p

De manera análoga y con las explicaciones y consideraciones dadas en el caso de las señales se toma la elección del conjunto de valores de p $[0, 2, 5, 10, 20, 40]$. Se ha contemplado el caso sin normalizar ($p = 0$) y varios valores de p para normalizar.

Las imágenes que se van a usar siguen siendo la de Lena como ejemplo en gris y la de la Alhambra como ejemplo en color. Se usa `m_term` con el objetivo de descartar un 80% de los coeficientes independientemente de p . Las gráficas halladas son:



(a) Gráfica para la imagen de Lena

(b) Gráfica para la imagen de la Alhambra

Imagen 3.26: Relación p - error

De nuevo para $p = 0$ (sin normalizar) es el peor error. El mejor es $p = 2$ y de ahí en adelante aumente. Al ser las imágenes un ejemplo más grande estas gráficas representan mucho mejor el efecto del normalizado en la aproximación greedy.

Aunque cuando p crece el error también parece como si hubiese una asíntota horizontal que acotase superiormente a la gráfica. Aunque computacionalmente se usa el valor $p = 0$ para los datos sin normalizar matemáticamente corresponde al valor $p = \infty$. Tomando el

criterio para ese p de $\frac{1}{p} = 0$ y $\frac{1}{q} = 1$ tenemos $2^{\frac{1}{q}} = 2$. Es probable que esa asíntota sea la marcada por la de los coeficientes sin normalizar.

Capítulo 4

Conclusiones y Trabajos Futuros

Como parte final del trabajo se van a exponer las diferentes conclusiones obtenidas a lo largo del desarrollo del mismo. Asimismo se dejan algunas propuestas y trabajos futuros relevantes para continuar con esta investigación.

4.1. Conclusiones

Las bases en el ámbito de espacios de Banach son un herramienta indispensable en Análisis Funcional. Hay varios tipos de bases y su estudio nos ha ayudado a comprender diversas propiedades de estos espacios y también a entender sus diferencias. Las bases greedy son útiles en la aproximación no lineal aunque generalmente poco implementables.

Sin embargo, algunas bases greedy como la de Haar se pueden adaptar a modelos prácticos. Se ha implementado una discretización de la misma para obtener un algoritmo de compresión de información eficiente y manejable. Los resultados son excelentes y se han optimizado los parámetros necesarios para realizar una compresión satisfactoria.

En el desarrollo del trabajo se han presentado resultados muy interesantes. De todos ellos me gustaría destacar las siguientes conclusiones:

- Una sucesión $(e_n)_{n=1}^{\infty}$ es una base de X si para cada $x \in X$ existe una única sucesión de escalares $(a_n)_{n=1}^{\infty}$ tal que $x = \sum_{n=1}^{\infty} a_n e_n$. Una sucesión $(e_n)_{n=1}^{\infty}$ es una base de Shauder si y solo si es una base para X . El sistema trigonométrico complejo $(1, e^{i\theta}, e^{-i\theta}, \dots)$ no es una base de $\mathcal{C}_{\mathbb{C}}(\mathbb{T})$.
- Hay una variedad de tipos de bases en espacios de Banach con propiedades y ca-

racterizaciones realmente interesantes. Aquí se han estudiado las bases monótonas, incondicionales, simétricas, acotadamente completas, shrinking y democráticas.

- La caracterización de Konyagin y Temlyakov: una base $\mathcal{B} = (e_n)_{n=1}^{\infty}$ de un espacio de Banach X es greedy si y solo si es incondicional y democrática.
- La conclusión matemática más importante para nuestros objetivos es que el sistema de Haar normalizado $\mathcal{H}_p = (h_n^p)_{n=1}^{\infty}$ es una base greedy en $L_p[0, 1]$ para $1 < p < \infty$.
- El algoritmo teórico de aproximación. Dado $f \in X$ y un diccionario \mathcal{D} en X , calculamos un funcional F_f tal que $\|F_f\|_{X^*} = 1$ y $F_f(f) = \|f\|_X$. Asumimos que existe $g_f \in \mathcal{D}$ de manera que $|F_f(g_f)| = \max_{g \in \mathcal{D}} |F_f(g)|$. Por último, una optimización más: encontrar el número a tal que $\|f - ag_f\|_X = \min_b \|f - bg_f\|_X$. Entonces:

$$G[\mathcal{D}, X](f) := ag_f, \quad R[\mathcal{D}, X](f) := f - ag_f.$$

- Se pueden discretizar las funciones de Haar obteniendo una transformada wavelet para la aproximación y compresión de arrays unidimensionales. Esto supone la construcción de un modelo implementable de aproximación greedy cuyo diseño es la base de una variedad aplicaciones de compresión de información.
- La primera aplicación de transformada de Haar discreta es sobre señales. Los algoritmos implementados resultan muy eficientes y las compresiones son satisfactorias. Se puede regular la calidad de la aproximación y la cantidad de información a almacenar. Es importante ajustar bien los parámetros para tener un compromiso entre estas dos medidas. Por ello, se han optimizado las relaciones que mantienen el umbral, el número de trozos y el espacio de normalización con el error.
- La última aplicación es de compresión de imágenes y por tanto se ha necesitado una extensión del diseño base al caso bidimensional. Las imágenes tienen un mayor peso que las señales por lo que la compresión resulta más necesaria y útil. Los resultados han sido muy satisfactorios pues con menos del 10% de los coeficientes el factor de compresión es altísimo mientras que la calidad sigue siendo excelente.

Al pintar los coeficientes greedy se observan las siluetas y bordes de los elementos de la imagen. La elección del umbral es importante para tener un equilibrio entre compresión y calidad de imagen y el óptimo se alcanza en el punto *knee* de la gráfica de la relación entre el porcentaje de descartados y error medio. El espacio de normalización $L_2[0, 1]$ es el mejor para los conjuntos de imágenes usados.

4.2. Trabajos futuros

La teoría de bases en espacios de Banach y la teoría de aproximación son campos de estudio con una gran potencia en la actualidad. Es por eso que aún se puede seguir investigando en cuestiones muy relacionadas a las aquí expuestas. Algunas pueden ser:

- La búsqueda de más aplicaciones de las bases y sucesiones básicas a la teoría isomórfica de espacios de Banach. Aunque esto es un mundo enorme, se proponen tres vías de investigación [3]:
 - 1) La estructura isomórfica de los espacios ℓ_p y c_0 .
 - 2) Subespacios complementados de ℓ_p ($1 \leq p < \infty$) y c_0 .
 - 3) Espacios isométricamente inyectivos.
- Estudio de nuevos tipos de bases como las Almost Greedy y Quasi-Greedy y también analizar nuevas caracterizaciones de las bases así como la relación que estas guardan con las que hemos estudiado [15].
- Otra tarea atrayente sería examinar nuevos ejemplos de bases greedy en espacios de Banach. Hacer pruebas con ejemplos sencillos y más visuales e incluso ver las aportaciones computacionales de otras bases greedy más complejas.
- Una propuesta verdaderamente atractiva de cara al futuro es el estudio de la relación que guardan las bases de un espacio con su espacio dual. Una primera cuestión a abordar sería: dada una base \mathcal{B} en un espacio de Banach X con alguna propiedad de tipo greedy, ¿qué podemos decir acerca de la sucesión de funcionales biortogonales $B^* = (e_n^*)_{n=1}^\infty$ en el espacio dual X^* ?
- Pensar nuevos ejemplos informáticos y ejecutar los algoritmos programados sobre éstos. En caso de que los algoritmos no sean útiles para los nuevos ejemplos analizar modificaciones que sí los harían válidos. Un buen comienzo sería compresión de audio y de vídeo.
- Realizar pruebas con un mayor número de señales o imágenes para poder establecer conclusiones más generales.
- Analizar el comportamiento de la elección del espacio de normalización $L_p[0, 1]$ en señales especiales. Dos ejemplos pueden ser señales con dientes de sierra y señales con mucha meseta.
- Estudio de otros métodos de compresión de información. Hay una diversidad de transformadas wavelet a parte de la de Haar, sería muy intrigante implementar y conocer otras transformadas y comparar sus resultados con los aquí obtenidos.

Bibliografía

- [1] J. LINDENSTRAUSS & L. TZAFIRI, *Classical Banach spaces I. Sequence spaces.* Ergebnisse der Mathematik und ihrer Grenzgebiete, Vol. 92. Springer-Verlag, Berlin-New York (1977).
- [2] J. LINDENSTRAUSS & L. TZAFIRI, *Classical Banach spaces II. Function spaces.* Ergebnisse der Mathematik und ihrer Grenzgebiete, Vol. 97. Springer-Verlag, Berlin-New York (1979).
- [3] F. ALBIAC & N. J. KALTON, *Topics in Banach Space Theory.* Graduate Texts in Mathematics (2015, Second Edition).
- [4] S. KARLIN, *Bases in Banach Space.* Duke Math J.. **15**, 971-985 (1948).
- [5] D. J. H. GARLING, *Symmetric bases of locally convex spaces.* Stud. Math. **30**, 163-181 (1968).
- [6] R. C. JAMES, *Bases and reflexivity of Banach spaces.* Ann. Math. (2) **52**, 518-527 (1950).
- [7] D. L. BURKHOLDER, *A nonlinear partial equation and the unconditional constant of the Haar system in L_p .* Bull. Am. Math. Soc. (N.S.) **7**(3), 591-595 (1982).
- [8] D. L. BURKHOLDER, *A proof of Pelczyński's conjecture for the Haar system.* Stud. Math. **91**(1), 79-83 (1988).
- [9] S. V. KONYAGIN & V. N. TEMLYAKOV, *A remark on greedy approximation in Banach spaces.* East J. Approx. **5**(3), 365-379 (1999).
- [10] V. N. TEMLYAKOV, *The best m -term approximation and greedy algorithms.* Adv. Comput. Math. **8**(3), 249-265 (1998).
- [11] V. TEMLYAKOV, *Nonlinear methods of approximation..* Found. Comput. Math. **3**, 33-107 (2003).
- [12] D. SALOMON, *Data compression. The Complete Reference.* Springer (2007, Fourth Edition).
- [13] S. MALLAT, *A Wavelet Tour of Signal Processing: The Sparse Way.* Academic Press

- (1998, Third Edition).
- [14] N. NIETO & D. M. OROZCO, *El uso de la transformada wavelet discreta en la reconstrucción de señales senosoidales*. Scientia et Technica Año XIV **38** (2008).
- [15] F. ALBIAC & J. ANSORENA & S. DILWORTH & D. KUTZAROVA, *Building highly conditional almost greedy and quasi-greedy bases in Banach spaces*. J. Funct. Anal **276**, 1893-1924 (2019).