

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE MATEMÁTICAS



Trabajo de Fin de Grado
Compressed Sensing para adquisición
sísmica

Francisco Javier Gutiérrez Oliet
Grado en Matemáticas
Curso 2020/21

Dirigido por:
Benjamin Ivorra
Prashanth Nadukandi

Capítulo 1

Introducción

Ante el desafío de reconstruir señales continuas partiendo de un conjunto de valores discretos, los métodos convencionales de muestreo usan como premisa el **Teorema de muestreo de Nyquist-Shannon** [11, 12]: la frecuencia de muestreo debe de ser superior al doble de la máxima frecuencia presente en la señal (la conocida como **tasa de Nyquist**). De hecho, es este mismo principio el que prevalece en casi la totalidad de los protocolos de recuperación de señales usados en campos muy diversos como lo son la electrónica o la medicina, por ejemplo.

Adicionalmente, en el campo de muestreo de datos, el procedimiento estándar de conversión de analógico a digital (normalmente representado por sus siglas en inglés *ADC*, que responden a *Analog-to-Digital Converter*) también hace uso de este teorema: la señal es muestreada uniformemente a la frecuencia que indica la tasa de Nyquist.

Sin embargo, existe una gran cantidad de situaciones en las que la obtención de una cantidad tan grande de valores discretos a partir de los cuales proceder a la reconstrucción se presenta como un problema inabordable [13].

En el presente trabajo se dará una descripción detallada de la teoría de *Compressed Sensing*, también llamada *Compressed Sampling* o por sus siglas *CS*, una novedosa técnica que rompe con las bases de la adquisición de datos, asegurando que, bajo ciertas premisas, podemos recuperar la totalidad de una señal partiendo de una muestra muy inferior a la que sugieren los métodos tradicionales. Para ello, *CS* se fundamenta en dos principios: la **dispersión** y la **incoherencia**.

Ya desde que Fourier publicó en 1822 su trabajo sobre la descomposición de señales como combinación de funciones sinusoidales [14], se pudo observar que muchas señales naturales con espectros muy complicados en el dominio del tiempo, presentaban una concentración en ciertos valores concretos al representarlos en el dominio de la frecuencia. Este hito permitió el desarrollo del concepto de dispersión: mediante la determinación de una referencia que permita representar una cierta señal con una mayor parte de coeficientes cercanos a cero, aseguramos que la mayor parte de la información que transmite inicialmente dicha señal es preservada tras comprimirla (con el objetivo de ahorrar memoria).

Por otro lado, la incoherencia permite asegurar que la geometría (y por tanto la dispersión) de la señal a comprimir se conserva al hacer una reducción de la dimensión del problema estudiado.

El trabajo está estructurado como sigue:

En el Capítulo 2 se proporciona un breve repaso de los fundamentos matemáticos sobre los que se sustenta la técnica, comenzando con la definición de espacio vectorial y llegando hasta el estudio de espacios de Hilbert, que será el tipo de espacios sobre el que se desarrollará la técnica.

El Capítulo 3 se dedica al estudio detallado del fenómeno de compresión: apoyándonos en transformadas de Fourier y ondículas mostraremos ejemplos sencillos de aplicaciones cotidianas que hacen uso de estas herramientas y presentaremos las técnicas más punteras que son habitualmente usadas para comprimir información.

Ya en el Capítulo 4 se presenta la técnica *CS*, haciendo hincapié en las ventajas que presenta frente a los métodos convencionales de muestreo y aportando ejemplos similares a los del capítulo anterior haciendo uso de la técnica.

En el Capítulo 5 se proporciona un ejemplo de su utilización en el campo de geosísmica, y más en concreto, de la optimización del proceso de adquisición de información acerca de la composición de las capas terrestres sobre la que se quiere plantear una posible extracción de petróleo.

Por último, el Capítulo 6 es una breve recapitulación de los resultados principales expuestos en el trabajo.

Dado que el objetivo del trabajo es la introducción a la técnica de *Compressed Sensing*, se han recopilado todos los archivos de código pertenecientes a los ejemplos presentados a lo largo del trabajo en el siguiente repositorio de GitHub:

https://github.com/fragut01/CS_tutorial.

Capítulo 2

Espacios de Señales

En este capítulo daremos una breve introducción a los espacios de señales y otros conceptos requeridos para entender los fundamentos de la técnica.

Es importante resaltar que, a lo largo de todo el trabajo, trataremos sin hacer referencia a ello señales continuas mediante una **discretización** de las mismas, con el objetivo de facilitar su manejo y aplicar la intuición y las herramientas geométricas de \mathbb{R}^3 , deseables para describir y comparar dos elementos. Además, este es el modo en el que trabajan los algoritmos de compresión que veremos en el Capítulo 3.

Por lo tanto, cada señal será un vector, y a su vez un elemento de un espacio vectorial.

Comenzamos por definir los conceptos principales de señal y espacio de señales:

Definición 2.0.1. Una **señal** es la representación de un valor en función de alguna magnitud como el tiempo, la frecuencia, etc. Cada señal quedará determinada por una letra minúscula x .

Definición 2.0.2. Un **espacio de señales** es un espacio vectorial cuyos elementos son señales al que nos referiremos por X .

Algunos ejemplos triviales de espacios de señales serían $X = \mathbb{R}$, $X = \mathbb{C}$ o $X = \mathbb{R}^n$, entre otros. Estaremos particularmente interesados en \mathbb{R}^n .

Definición 2.0.3. Denotaremos por $L_2[a, b]$ al espacio de señales que contiene a todas las señales $x(t)$ tal que $\int_a^b |x(t)|^2 dt < \infty$.

A modo de recordatorio, presentamos el concepto de espacio vectorial, en el que induciremos una norma dando lugar a los espacios normados. Seguidamente, veremos que en cualquier espacio normado se puede definir de manera consistente la distancia entre dos puntos, motivando así la noción general de producto escalar y **ortogonalidad**, que será clave en el posterior desarrollo de la teoría de *Compressed Sensing*.

2.1. Espacios vectoriales

Definición 2.1.1. Un espacio vectorial $(X, \mathbb{K}, +, \cdot)$ está formado por un conjunto no vacío X , un cuerpo \mathbb{K} y dos operaciones para las cuales será cerrado: una operación suma $+ : X \times X \rightarrow X$, y una operación producto $\cdot : \mathbb{K} \times X \rightarrow X$, de manera que:

- $(X, +)$ forma un grupo abeliano:
 - (a) $\forall x, y \in X, x + y = y + x$ (comutatividad).
 - (b) $\forall x, y, z \in X, (x + y) + z = x + (y + z)$ (asociatividad).
 - (c) $\exists 0 \in X$ tal que $0 + x = x, \forall x \in X$ (existencia de elemento neutro).
 - (d) $\forall x \in X, \exists y = -x \in X$ tal que $x + y = 0$ (existencia de elemento inverso).
- (X, \mathbb{K}, \cdot) cumple las siguientes propiedades:
 - (a) $\forall a, b \in \mathbb{K}, \forall x \in X, a \cdot (b \cdot x) = (a \cdot b) \cdot x$, (asociatividad).
 - (b) $1 \cdot x = x, \forall x \in X$, donde 1 es la unidad para el producto en \mathbb{K} .
 - (c) $\forall a, b \in \mathbb{K}, \forall x \in X, (a + b) \cdot x = a \cdot x + b \cdot x$ (distributiva respecto de la suma escalar).
 - (d) $\forall a \in \mathbb{K}, \forall x, y \in X, a \cdot (x + y) = a \cdot x + a \cdot y$ (distributiva respecto de la suma vectorial).

Observación 2.1.2. Generalmente nos referiremos al espacio vectorial $(X, \mathbb{K}, +, \cdot)$ por su conjunto X .

Observación 2.1.3. Como ya hemos comentado, X será un espacio de señales y \mathbb{K} será \mathbb{R} .

2.1.1. Espacios normados

Definición 2.1.4. Una **norma** en un espacio vectorial X es una aplicación $\|\cdot\| : X \rightarrow \mathbb{R}$, que a cada vector x le hace corresponder un numero real $\|x\|$, verificando:

- (a) $\|\lambda x\| = |\lambda| \|x\|, \forall x \in X, \forall \lambda \in \mathbb{R}$ (homogeneidad por homotecias).
- (b) $\|x\| = 0 \Leftrightarrow x = 0, \forall x \in X$ (no degeneración).
- (c) $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in X$ (desigualdad triangular).

Un **espacio normado** es un par $(X, \|\cdot\|)$ formado por un espacio vectorial X en el que hemos inducido una norma $\|\cdot\|$.

Definición 2.1.5. Una aplicación $\|\cdot\|$ es una **cuasinorma** si cumple todas las condiciones de la norma a excepción de la desigualdad triangular, que es reemplazada por la siguiente:

$$\forall x, y \in X, \exists K \in (0, 1) : \|x + y\| \leq K(\|x\| + \|y\|)$$

Observación 2.1.6. Del mismo modo que en los espacios vectoriales, cuando no haya lugar a confusión nos referiremos al espacio normado $(X, \|\cdot\|)$ nombrando al conjunto X .

Observación 2.1.7. Aunque al número real $\|x\|$ lo llamaremos también *norma del vector* x , no debe confundirse con la *aplicación* norma definida en el espacio vectorial.

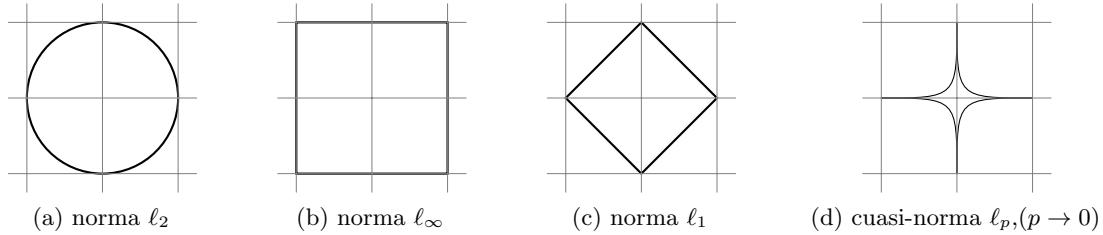
Proposición 2.1.8. Para todo vector $x \in X$ no nulo, $\|x\| > 0$.

Demostración. Se deduce de las propiedades de la norma, puesto que:

$$\|-x\| = |-1| \|x\| = \|x\| \Rightarrow 0 = \|x + (-x)\| \leq \|x\| + \|-x\| = 2\|x\|,$$

por lo que $\|x\| > 0, \forall x \neq 0$. □

Figura 2.1: Circunferencia unidad en \mathbb{R}^2 resultante de inducir distintas ℓ_p .



Ejemplo 2.1.9. La norma más habitual es la conocida como **norma euclídea**, que inducida sobre $X = \mathbb{R}^n$, se expresa como sigue:

$$\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$$

Observación 2.1.10. En el caso de señales **discretas**, correspondientes a espacios de **dimensión finita**, podemos pensar en las señales como vectores en un espacio euclídeo n-dimensional, que denotamos por \mathbb{R}^n .

Ejemplo 2.1.11. Además de la euclídea, los ejemplos más notables de normas en \mathbb{R}^n , ($n > 1$) son las conocidas como **normas ℓ_p** , con $p > 1$. Para un vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ dado, vienen determinadas por:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (2.1)$$

si $p \in (1, \infty)$, y por $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$ para $p = \infty$.

Observación 2.1.12. Será de utilidad extender la noción de normas ℓ_p al caso $p < 1$. En este caso, la aplicación definida por (2.1) es una **cuasinorma**.

Tambiénaremos frecuentemente uso de la notación $\|x\|_0 := |\text{supp}(x)|$, donde $\text{supp}(x) := \{i : x_i \neq 0\}$ es el soporte del vector x , por lo que la “norma” ℓ_0 nos permitirá identificar las componentes no nulas de un vector dado, concepto fundamental cuando hablemos del principio de dispersión (Sección 3.1). En la Figura 2.1 se presenta la circunferencia unidad \mathbb{S}^1 con diversos tipos de norma inducida¹.

Observación 2.1.13. Nótese que ℓ_0 formalmente no es ni siquiera una cuasinorma, no obstante, el abuso de notación queda justificado por el hecho de que se verifique la siguiente igualdad:

$$\|x\|_0 = |\text{supp}(x)| = \lim_{p \rightarrow 0} \|x\|_p^p$$

motivo por el que nos referiremos a ella como “norma ℓ_0 ”.

Aunque las normas presentadas en el ejemplo anterior sean distintas que la euclídea, veremos que toda norma inducida en \mathbb{R}^n es equivalente. Para ello, primero debemos definir formalmente lo que entendemos por normas equivalentes:

Definición 2.1.14. Dos normas sobre un mismo espacio vectorial X se dicen **equivalentes** cuando inducen la misma topología sobre X .

¹Recordamos que la circunferencia unidad en \mathbb{R}^2 con una cierta norma $\|\cdot\|$ inducida está formada por $\{(x, y) \in \mathbb{R}^2 : \|(x, y)\| = 1\}$

La siguiente proposición nos aporta un método para identificar equivalencias entre normas:

Proposición 2.1.15. Dos normas $\|\cdot\|$ y $\|\cdot\|^*$ sobre X son equivalentes si y sólo si existen dos constantes $A, B > 0$ tales que

$$A \|x\| \leq \|x\|^* \leq B \|x\|, \forall x \in X \quad (2.2)$$

Remark. En espacios normados de **dimensión finita**, todas las normas son equivalentes. Esto quiere decir que siempre es posible encontrar las constantes A y B de la Proposición 2.1.15. No obstante, su cálculo no tiene por qué ser sencillo, motivo por el cual es importante elegir las normas de manera conveniente.

Definición 2.1.16. Un espacio normado $(X, \|\cdot\|)$ se dice que es un **espacio de Banach** si es **completo**, es decir, si toda sucesión de Cauchy² es convergente a un elemento **perteneciente a X** .

Mientras que las normas son las encargadas de evaluar la importancia de una cierta señal, las distancias y las métricas ayudan a saber cómo de diferentes son dos señales dadas.

2.1.2. Espacios métricos

Definición 2.1.17. Dado un conjunto no vacío E , una **métrica** $d : E \times E \rightarrow \mathbb{R}$ es una función que cumple las siguientes propiedades:

- (a) $d(a, b) = d(b, a)$ (simétrica)
- (b) $d(a, b) = 0 \Leftrightarrow a = b$ (no degenerativa).
- (c) $d(a, c) \leq d(a, b) + d(b, c)$ (desigualdad triangular).

Si d es una métrica en E , diremos que $d(a, b)$ es la **distancia entre a y b**, y el par (E, d) recibirá el nombre de **espacio métrico**.

Observación 2.1.18. Nótese que un conjunto E puede tener múltiples métricas definidas, que darán lugar a espacios métricos diferentes.

Definición 2.1.19. Dado un espacio normado $(X, \|\cdot\|)$, se determina la **distancia** entre dos elementos de X como:

$$d(x, y) := \|y - x\|, \forall x, y \in X$$

Observación 2.1.20. En el caso de espacios vectoriales, una métrica como en la Definición 2.1.17 siempre tiene asociada una norma, y de hecho esta norma puede ser recuperada mediante el cálculo de:

$$d(0, x) := \|x\|, \forall x \in X$$

Por ello, consideraremos todo espacio normado como un espacio métrico, con la métrica (o distancia) asociada a su norma.

Observación 2.1.21. El hecho de que todo espacio normado pueda verse como un espacio métrico no implica que toda métrica provenga de una norma. Por ejemplo, sea E un conjunto no vacío y $\delta : E \times E \rightarrow \mathbb{R}$ la métrica discreta definida para cada par de elementos x, y de E como:

$$\delta(x, y) = \begin{cases} 0, & \text{si } x = y \\ 1, & \text{si } x \neq y \end{cases}$$

²Recordamos que una sucesión (x_n) es de Cauchy $\Leftrightarrow \forall \varepsilon > 0, \exists N \in \mathbb{N}$ tal que si $m, n > N \Rightarrow \|x_n - x_m\| < \varepsilon$

que es una métrica como se comprueba fácilmente. No obstante, si δ tuviera una norma asociada, entonces, como hemos visto en la Observación 2.1.20, $\forall x \neq 0$, $\|x\| = \delta(x, 0) = 1$, y también $\|2x\| = \delta(2x, 0) = 1$, lo que contradice la propiedad (a) de las normas puesto que estaríamos afirmando que $\|\lambda x\| = \|x\|$, para $\lambda = 2$.

Definición 2.1.22. Si el espacio métrico (X, d) proviene de una norma y es además completo, entonces X es un **espacio de Banach**.

Ejemplo 2.1.23. A continuación, presentamos algunos ejemplos típicos de métricas en $X = \mathbb{R}^n$. Nótese que son métricas asociadas a las normas ya presentadas en la sección anterior:

- $d_2(x, y) := (\sum_{i=1}^n |x_i - y_i|^2)^{1/2}$, conocida como métrica euclídea.
- $d_\infty(x, y) := \max_{i=1, \dots, n} |y_i - x_i|$
- $d_1(x, y) := \sum_{i=1}^n |y_i - x_i|$

Observación 2.1.24. Los espacios métricos (\mathbb{R}^n, d_2) son completos, y por lo tanto espacios de Banach, para todo $n \in \mathbb{N}$.

Por otro lado, el espacio (\mathbb{Q}, d_1) es un ejemplo típico de espacio métrico que **no** es completo.

2.1.3. Producto escalar y espacios de Hilbert

Ante la posibilidad de que dos señales con misma norma y métrica presenten un comportamiento muy dispar, introducimos el concepto de producto escalar, necesario para hablar de ortogonalidad. Tomaremos como espacio de llegada $\mathbb{K} = \mathbb{R}$ por ser el espacio con el que trabajaremos.

Definición 2.1.25. Un **producto escalar** definido en un \mathbb{R} -espacio vectorial es una aplicación $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{R}$ verificando las siguientes propiedades:

- (a) $\langle x, y \rangle = \langle y, x \rangle$, $\forall x, y \in X$ (simétrica).
- (b) $\langle ax + by, z \rangle = a\langle x, z \rangle + b\langle y, z \rangle$; $\langle x, cy + dz \rangle = c\langle x, y \rangle + d\langle x, z \rangle$ $\forall x, y, z \in X$, $\forall a, b, c, d \in \mathbb{R}$ (bilineal).
- (c) $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0 \Leftrightarrow x = 0$, $\forall x \in X$ (definida positiva).

Del mismo modo que toda norma induce una métrica, todo producto escalar induce de manera natural una norma:

$$\|x\|_i = \sqrt{\langle x, x \rangle} \quad (2.3)$$

Cuando el espacio métrico resultante de inducir la norma (2.3) en un espacio vectorial sea completo, diremos que el espacio considerado es un **espacio de Hilbert**³.

Ejemplo 2.1.26. Consideremos $X = \mathbb{R}^n$ con el producto escalar definido por

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i = y^t x$$

La norma inducida corresponde a $\|x\|_i = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^n x_i^2} = \|x\|_2$, la norma euclídea.

Puesto que además (\mathbb{R}^n, d_2) es completo, concluimos que es un espacio de Hilbert (y también un espacio de Banach).

³Nótese que todo espacio de Hilbert es siempre un espacio de Banach porque, por definición, un espacio de Hilbert es completo con respecto a la norma asociada a su producto escalar. El recíproco en general no es cierto; Una condición necesaria y suficiente para que un espacio de Banach sea también un espacio de Hilbert es que se verifique la identidad del paralelogramo.

Estamos ya en posición de introducir el concepto de ortogonalidad, que jugará un papel fundamental a la hora de clasificar y representar señales:

Definición 2.1.27. Dos vectores $x, y \in X$ son **ortogonales** si el producto escalar $\langle x, y \rangle = 0$, en cuyo caso lo denotamos por $x \perp y$. Se sigue que un conjunto de vectores $S \subseteq X$ es ortogonal si $\langle x, y \rangle = 0, \forall x, y \in S$.

Definición 2.1.28. Un conjunto de vectores $S \subseteq X$ es **ortonormal** si es ortogonal y además $\|s\| = \sqrt{\langle s, s \rangle} = 1, \forall s \in S$, donde $\|\cdot\|$ es la norma inducida correspondiente.

2.2. Operadores lineales

Definición 2.2.1. Sean E y F espacios normados y sea T una aplicación de E en F . Llamaremos a T un **operador** de E en F . Indicaremos por $T(x) = Tx$ a la imágen de x por T .

Decimos que T es **lineal** si

$$T(\lambda x + \mu y) = \lambda Tx + \mu Ty$$

para todo $x, y \in \mathbb{R}$.

Observación 2.2.2. Obsérvese que $T0 = 0$, ya que

$$T0 = T(2 \cdot 0) = 2T0 \Rightarrow T0 = 0.$$

Definición 2.2.3. Si $T(E) = \{Tx : x \in E\} = F$ decimos que T es **sobreyectivo**. Si siempre que $Tx = Ty$ se tiene que $x = y$, entonces diremos que T es **inyectivo**.

Proposición 2.2.4. Dado $T : E \rightarrow F$ un operador lineal, T es inyectivo $\Leftrightarrow Tx = 0$ implica que $x = 0$.

Demostración. \Rightarrow] Si $Tx = 0$, entonces $Tx = T0$, luego la inyectividad implica que $x = 0$.

\Leftarrow] Si $Tx = Ty$, se tiene que $T(x - y) = 0$, y por la hipótesis deducimos que $x = y$. \square

Definición 2.2.5. El operador $T : E \rightarrow F$ se dice **continuo** en x si $\forall \varepsilon > 0$, existe $\delta > 0$ tal que siempre que $\|y - x\|_E \leq \delta$ se tiene que $\|Ty - Tx\|_F \leq \varepsilon$. Esto equivale a que siempre que (x_n) sea una sucesión de E convergente a x se tiene que $Tx_n \rightarrow Tx$.

Proposición 2.2.6. El operador $T : E \rightarrow F$ es continuo en $E \Leftrightarrow$ es continuo en un punto de E .

Demostración. \Leftarrow] Supongamos que T es continuo en z . Sea $x \in E$ arbitrario y $((x_n) \rightarrow x)$. Entonces, $(x_n - x + z) \rightarrow z$. Luego $T(x_n - x + z) \rightarrow Tz$. Como $T(x_n - x + z) = Tx_n - Tx + Tz$, se sigue que $Tx_n \rightarrow Tx$. \square

Definición 2.2.7. Un conjunto $A \subset E$ se dice **acotado** si existe $M > 0$ tal que $\|x\| \leq M, \forall x \in A$. Esto equivale a que A esté contenido en la bola cerrada de centro 0 y radio M : $\overline{B}(0, M)$.

Definición 2.2.8. Un operador lineal $T : E \rightarrow F$ se dice **acotado** si existe $M \geq 0$ tal que

$$\|Tx\|_F \leq M \|x\|_E, \text{ para todo } x \in E. \quad (2.4)$$

Denotaremos por $\mathcal{L}(E, F)$ al espacio de todos los operadores lineales y acotados de E en F .

Proposición 2.2.9. Sean E, F espacios normados y sea $T : E \rightarrow F$ operador lineal. Entonces T es continuo en $E \Leftrightarrow T$ es acotado.

Demostración. [\Leftarrow] Si T es acotado y $(x_n) \rightarrow 0$, entonces $Tx_n \rightarrow 0$ ya que

$$\|Tx_n\|_F \leq \|T\| \|x_n\| \rightarrow 0$$

luego T es continua en 0 y así en E .

[\Rightarrow] Procedemos por contradicción: si T no fuera acotado, entonces para todo $n \in \mathbb{N}$ existe $x_n \in E$ tal que $\|Tx_n\|_F > n \|x_n\|_E$. Sea $y_n = (1/n \|x_n\|)x_n$. Entonces $\|y_n\| = 1/n \rightarrow 0$, luego $y_n \rightarrow 0$ pero $\|Ty_n\|_F = \|Tx_n\|_F / n \|x_n\|_E > 1$ para todo $n \in \mathbb{N}$. Así Ty_n no converge a 0, lo que contradice la continuidad de T en 0. \square

Proposición 2.2.10. Si $T \in \mathcal{L}(E, F)$, se define

$$\|T\| = \min \{M : M \text{ satisface (2.4)}\} \quad (2.5)$$

Se tiene que $(\mathcal{L}(E, F), \|\cdot\|)$, donde $\|\cdot\|$ es la norma (2.5) es un espacio normado. Si además F es un espacio de Banach, entonces $(\mathcal{L}(E, F), \|\cdot\|)$ es un espacio de Banach.

Demostración. Se puede encontrar en cualquier texto de introducción al análisis funcional, como por ejemplo [1]. \square

2.3. Bases y *frames*

Una vez definidos estos conceptos básicos, pasamos a estudiar comportamientos particulares de los espacios normados de dimensión finita y formas de caracterizarlos:

Definición 2.3.1. Sea E un espacio vectorial sobre \mathbb{K} . Decimos que E es de **dimensión finita** si existe un $n \in \mathbb{N}$ y $\{e_1, \dots, e_n\}$ vectores en E de manera que para cada $x \in E$, existen $\lambda_1, \dots, \lambda_n \in \mathbb{K}$ tales que $x = \sum_{j=0}^n \lambda_j e_j$, en cuyo caso se dice que $\{e_1, \dots, e_n\}$ es un **sistema de generadores** de E .

Al menor natural n tal que existe un sistema de generadores de n elementos se le denomina **dimensión** de E y se denota por $\dim(E)$. La determinación de un sistema de generadores constituido únicamente por esos n elementos motiva la siguiente subsección:

2.3.1. Bases

Definición 2.3.2. A un sistema de generadores con tantos elementos como la dimensión de E se le llama **base** de E .

Definición 2.3.3. Dada una base $\mathcal{E} = \{e_1, \dots, e_n\}$ y un $x \in E$, a la n -úpla $(\lambda_1, \dots, \lambda_n) \in \mathbb{K}$ tal que $x = \sum_{j=0}^n \lambda_j e_j$ se le llama **coordenadas** de x en la base \mathcal{E} .

Observación 2.3.4. Es fácil comprobar que la base de un espacio E no es única y que cada base nos dará unas coordenadas distintas para un vector $x \in E$. Además, las coordenadas de un elemento $x \in E$ son únicas fijada una base de E , hecho que será primordial cuando hablemos de la robustez que aportan los *frames* en esta misma sección.

Estamos en posición de fijar como espacio de señales (discreto) el espacio de Hilbert real de dimensión finita $X = \mathbb{R}^n$. En él, cada señal quedará representada por un vector x de manera que:

$$x = \sum_{i=1}^n a_i \psi_i \quad (2.6)$$

donde la familia $\{\psi_i\}_{i=1}^n$ está formada por los vectores que constituyen una base de nuestro espacio de señales \mathbb{R}^n y las valores de las coordenadas a_i no están determinados a priori.

Definición 2.3.5. Apoyándonos en la Subsección 2.1.3, una cierta base es **ortonormal** si cumple que $\langle \psi_i, \psi_j \rangle = 0$, si $i \neq j$ y $\langle \psi_i, \psi_i \rangle = 1$. Bajo estas condiciones, la matriz de la base dual coincide con la matriz transpuesta de nuestra base.

Como consecuencia de este último resultado se tiene una propiedad fundamental de las bases ortonormales:

Proposición 2.3.6. Sea $x \in \mathbb{R}^n$ una señal, $\Psi = \{\psi_i\}_{i=1}^n$ una base ortonormal del espacio \mathbb{R}^n . Entonces, las coordenadas a_i de x respecto de Ψ se pueden obtener mediante el cálculo de:

$$a_i = \langle x, \psi_i \rangle,$$

de donde se sigue que la Ecuación 2.6 coincide con:

$$x = \sum_{i=1}^n \langle x, \psi_i \rangle \psi_i \quad (2.7)$$

Por este motivo, trabajaremos siempre con bases ortonormales para identificar nuestras señales, en cuyo caso quedan bien definidos los siguientes operadores lineales asociados:

- **Operador síntesis Ψ .** Es el operador cuya matriz contiene como columnas los vectores de la base Ψ , el cual reproduce una señal al ser operado con un vector de coeficientes cualquiera $\alpha \in \mathbb{R}^n$

$$x = \Psi \alpha$$

- **Operador análisis Ψ^T .** Es el operador cuya matriz coincide con la de la base dual asociada a Ψ , que al ser operado con una cierta señal x , devuelve una secuencia de coeficientes de expansión α .

$$\alpha = \Psi^T x$$

Definición 2.3.7. Denotaremos por Ψ a la **matriz de representación** de la base Ψ : la matriz cuyas columnas son los vectores que forman parte de la base Ψ .

Nótese que, gracias a las buenas propiedades de las bases ortonormales, se cumple la ecuación:

$$\Psi^{-1} = \Psi^T, \quad (2.8)$$

Será de utilidad usar un tipo especial de sistemas de generadores, conocidos como *frames*:

2.3.2. Frames

Definición 2.3.8. Un *frame* es una sistema de generadores $\{\Psi\}_{i=1}^d$ definido en \mathbb{R}^n con $n \leq d$ que tiene como matriz de representación $\Psi \in \mathbb{R}^{n \times d}$ de manera que para todo $x \in \mathbb{R}^n$:

$$A \|x\|_2^2 \leq \|\Psi^T x\|_2^2 \leq B \|x\|_2^2 \quad (2.9)$$

para ciertos A, B cumpliendo que $0 < A \leq B < \infty$.

Observación 2.3.9. Nótese que la condición $A > 0$ implica que las filas de Ψ son linealmente independientes.

Definición 2.3.10. Ψ recibe el nombre de *Frame de Parseval*⁴ si la condición (2.9) se cumple para $A = B = 1$, es decir, si se tiene que:

$$\|x\|_2^2 = \|\Psi^T x\|_2^2 \quad (2.10)$$

Interpretando la identidad desde el punto de vista del procesamiento de la señal, (2.10) implica que la energía de la señal x es preservada al ser representada con el *frame* Ψ [2].

Lema 2.3.11. Toda base ortonormal $(e_i)_{i=1}^n$ de \mathbb{R}^n es un *Frame de Parseval*.

Ejemplo 2.3.12. A partir de $(e_i)_{i=1}^n$ una base ortonormal cualquiera de \mathbb{R}^n siempre podemos construir un *Frame de Parseval* como sigue:

(a) El sistema de generadores:

$$\Psi = (e_1, 0, e_2, 0, \dots, e_n, 0)$$

es un *Frame de Parseval*, que como se aprecia puede contener vectores nulos.

(b) El sistema de generadores:

$$\widehat{\Psi} = (e_1, e_2/\sqrt{2}, e_2/\sqrt{2}, e_3/\sqrt{3}, e_3/\sqrt{3}, e_3/\sqrt{3}, \dots, e_n/\sqrt{n}, \dots, e_n/\sqrt{n})$$

es un *Frame de Parseval*, que puede contener varias veces al mismo vector, e incluso contener vectores cuyas normas convergan a cero.

Proporcionamos a continuación un lema que permite identificar *Frames de Parseval* de manera sencilla para el caso en que $X = \mathbb{R}^n$:

Lema 2.3.13. Una familia finita de vectores $\{\psi_i\}_{i=1}^d$ en \mathbb{R}^n con $n < d$ es un *Frame de Parseval* si y sólo si las filas de la matriz de representación Ψ correspondiente son vectores ortonormales vistos como elementos de \mathbb{R}^d .

Ejemplo 2.3.14. Con el objetivo de clarificar este último resultado, se presenta para $X = \mathbb{R}^2$ un *Frame de Parseval* que se conoce como *Frame Mercedes-Benz* (ver Figura 2.2) formado por $d = 3$ elementos. En concreto, está formado por el sistema de generadores $\{\varphi_1, \varphi_2, \varphi_3\}$ de manera que:

$$\Psi = \begin{pmatrix} 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ 1 & -1/2 & -1/2 \end{pmatrix} = \begin{pmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{pmatrix}$$

Puesto que las filas son linealmente independientes entendidos como elementos de \mathbb{R}^3 , concluimos que Ψ es un *Frame de Parseval* de \mathbb{R}^2 . Efectivamente, se tiene que:

$$\|\Psi^T x\|_2^2 = \sum_{i=1}^3 |\langle x, \varphi_i \rangle|^2 = \|x\|_2^2$$

⁴El Identidad de Parseval es una generalización del Teorema de Pitágoras para espacios de Hilbert. Nos permite asegurar que las distancias y las propiedades geométricas de la señal original se conservan al representarla con un cierto *Frame de Parseval* Ψ

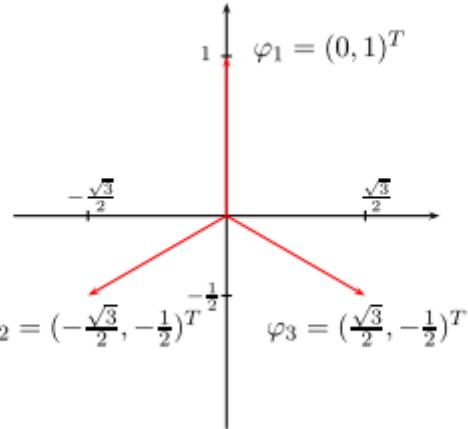


Figura 2.2: Frame Mercedes-Benz.

Proposición 2.3.15. En el caso de los espacios $X = \mathbb{R}^n$ los valores de A y B se corresponden, respectivamente, con el menor y el mayor autovalor de $\Psi\Psi^T$.

Observación 2.3.16. Para una cierta señal x , existe una infinidad de vectores de coeficientes α que satisfacen la ecuación

$$x = \Psi\alpha$$

En particular, cada elección de un *frame* dual $\widehat{\Psi}$ aporta un vector de coeficientes diferente, en adelante α .

Definición 2.3.17. $\widehat{\Psi}$ es **frame dual alternado** asociado a Ψ si satisface la ecuación:

$$\Psi\widehat{\Psi}^T = \widehat{\Psi}\Psi^T = I_n,$$

donde I es la matriz identidad de dimensión $n \times n$.

Si además $\widehat{\Psi}$ cumple que $\widehat{\Psi} = (\Psi\Psi^T)^{-1}\Psi$ entonces es el **frame dual canónico**.

Observación 2.3.18. Al ser las filas de Ψ linealmente independientes, podemos asegurar que $\Psi\Psi^T$ es invertible, y por lo tanto que $\widehat{\Psi}$ está bien definido.

Presentamos adicionalmente la siguiente proposición por su interés en secciones posteriores. Demostración de esta y resultados más en detalle sobre *frames* pueden encontrarse en [3]

Proposición 2.3.19. La elección del vector de coeficientes α_1 como

$$\alpha_1 = \Psi^T (\Psi\Psi^T)^{-1} x$$

minimiza la norma euclídea. Es decir, se tiene que $\|\alpha_1\|_2 \leq \|\alpha\|_2$, para todo α tal que $x = \Psi\alpha$.

De hecho, esta α_1 es conocida como la solución siguiendo el método de *frames*, o *the Method of Frames (MoF) solution*.

Observación 2.3.20. El concepto de *frame* es fundamental en el desarrollo de la compresión de señales puesto que permite aportar representaciones mas robustas de un cierto espacio de señales

apoyándose en la **redundancia**: la representación de una cierta señal x respecto del *frame* Ψ **no** es única. Gracias a las múltiples combinaciones de coeficientes que permiten representar x , podemos preservar señales bajo condiciones de ruido y evitar pérdidas en general.

Puesto que los resultados a los que queremos llegar tienen la computación como ingrediente esencial, hay que tener presente que las aplicaciones de herramientas y métodos teóricos a problemas prácticos reales presentan imprecisiones y limitaciones.

Dado que, como no puede ser de otra manera en el caso computacional, estaremos tratando con discretizaciones de señales continuas, estará siempre presente el conocido como **error de cuantización**: al discretizar la recta real, el ordenador crea una relación de equivalencia en \mathbb{R} , inducida por la partición que hace de la recta. Por lo tanto, cada clase está compuesta por los números pertenecientes a un intervalo, cuyo representante es el punto medio de dicho intervalo. Evidentemente, este hecho es fuente de imprecisiones y errores y el coste computacional crece conforme se intenta reducir la longitud de cada clase con el objetivo de ganar precisión.

Aquí es donde la utilización de frames puede resultar provechosa: al aumentar la redundancia del frame (es decir, al añadir mas elementos que sean combinación lineal de los ya pertenecientes a una base) se reduce el error cometido en la recuperación de la señal [4]. Lo interesante de este resultado es que aporta un método para reducir el error de reconstrucción distinto del más obvio, que sería ir reduciendo el error de cuantización de las muestras. Por tanto, cuando resulte menos costoso aumentar la redundancia del frame con el que se trabaja que reducir los errores de cuantización, los beneficios que aporta el uso de frames son considerables.

Presentamos por último la elaboración del producto tensorial de dos matrices, que será de utilidad para operar los espacios de representación de la señal de interés una vez conocidas las matrices respecto de un cierto frame o base apropiados.

2.4. Producto de Kronecker de dos matrices

Sean $A \in \mathbb{R}^{m \times n}$ y $B \in \mathbb{R}^{p \times q}$. Definimos el **producto de Kronecker** de A y B como:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{nn}B \end{bmatrix} \in \mathbb{R}^{mp \times nq} \quad (2.11)$$

donde cada término a_{ij} es el elemento de la matriz A situado en la fila i y la columna j .

Observación 2.4.1. Aunque al igual que el producto usual de matrices sigue sin ser comutativo, el producto de Kronecker de dos matrices **no** queda reservado a matrices entre las que exista una determinada compatibilidad entre las dimensiones, como muestra el siguiente ejemplo en el que calculamos el producto de Kronecker de una matriz A de dimensiones 2×3 con una B de dimensiones 2×2 :

Ejemplo 2.4.2. Sean $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}$ y $B = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$ dos matrices. Se tiene que

$$A \otimes B = \begin{bmatrix} B & 2B & 3B \\ 3B & 2B & B \end{bmatrix} = \begin{bmatrix} 2 & 1 & 4 & 2 & 6 & 3 \\ 2 & 3 & 4 & 6 & 6 & 9 \\ 6 & 3 & 4 & 2 & 2 & 1 \\ 6 & 9 & 4 & 6 & 2 & 3 \end{bmatrix} \quad (2.12)$$

Para poder explicar el motivo por el que requerimos de esta operación, debemos de definir primero el siguiente tipo de vector:

Definición 2.4.3. Sea $A \in \mathbb{R}^{m \times n}$ una matriz y denotemos por a_i a la columna i -ésima, de manera que podemos reescribir $A = [a_1 a_2 \dots a_n]$. Definimos entonces el vector $\text{vec}(A)$ como el vector nm -dimensional resultante de apilar las columnas de A consecutivamente, es decir:

$$\text{vec}(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

Proposición 2.4.4. Para cualesquiera tres matrices A, B, C para las que esté definido el producto usual de matrices ABC , se verifica la siguiente igualdad:

$$\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B) \quad (2.13)$$

Demostración. Ver [15]. □

Capítulo 3

Dispersión y compresión de señales

Como se ha visto en el capítulo anterior, la representación de una señal apoyándose en *frames* y bases ortonormales proporciona robustez y ayuda a conservar más precisamente la información de la señal en condiciones de ruido, a la vez que preserva la geometría intrínseca de la señal. Procedemos ahora a introducir el concepto de dispersión, que es el principio sobre el que se sustentan los métodos de compresión de datos que usamos en nuestro día a día.

3.1. Dispersión

Cuando hablamos del principio de **dispersión** o *sparsity* en inglés, nos referimos al hecho de explotar que gran parte de las señales naturales que podríamos estar interesados en muestrear, pueden ser expresadas en una base o *frame* conveniente de modo que gran parte de sus coordenadas sean muy cercanas a cero¹.

Esto permite que podamos percibir los coeficientes pequeños indistintamente como nulos sin que esto suponga una pérdida significativa en la recepción de información que la señal debe transmitir, reduciendo la memoria necesaria para almacenar la señal.

Definición 3.1.1. Una señal $x \in \mathbb{R}^n$ es **K-dispersa** cuando su vector de coeficientes α asociado a alguna base o *frame* Ψ tiene exactamente K entradas no nulas, es decir, cuando:

$$\|\alpha\|_0 = K,$$

donde $\|\cdot\|_0$ es la "norma" ℓ_0 que se definió en la Observación 2.1.12.

Observación 3.1.2. A nivel práctico, extenderemos el concepto de señal K-dispersa a señales que respecto de cierto *frame* poseen una representación donde todos menos los K mayores coeficientes son **muy próximos** a cero, puesto que al desestimarlos (con el objetivo de comprimir la señal) aseguramos que la información fundamental sobre x está siendo preservada en los coeficientes de mayor valor.

¹Para ayudar a comprender lo inabarcable que es el espacio de señales, podemos recurrir a [16], en el que Borges ya emula una situación en la que, de todos los posibles libros que podrían ser escritos algún día, los que contienen un mensaje coherente y bien redactado ocupan una mínima porción. Del mismo modo, en el vasto espacio de imágenes, compuesto en su mayoría por ruido aleatorio, el conjunto formado por imágenes naturales que podríamos estar interesados en comprimir, tiene un tamaño ínfimo.

Este es el principio fundamental sobre el que se basa la compresión de datos: mediante el uso de bases y *frames* universales que permitan representar una gran variedad de señales de manera muy dispersa, requerimos de menos memoria para preservar dichas señales al registrar un menor número de coeficientes.

Ya desde que Fourier publicó en 1822 su trabajo sobre la descomposición de señales como combinación de funciones sinusoidales [14], se pudo observar que muchas señales naturales con espectros muy complicados en el dominio del tiempo, presentaban una concentración en ciertos valores concretos al representarlos en el dominio de la frecuencia.

Casi un siglo más tarde, en 1909, Alfred Haar publicó un artículo [17] en el que presentaba una secuencia de funciones que proporcionaba un sistema numerable ortonormal para el espacio $L^2[0, 1]$, que como vimos en la Definición 2.0.3 es el conjunto formado por todas las funciones de cuadrado integrable en el intervalo $[0, 1]$. Aunque hoy en día a este concepto se le conoce como **ondícula de Haar**, conocida por ser la ondícula más simple, el desarrollo del concepto de **ondícula o Wavelet** no ocurrió hasta mediados del siglo XX y fue gracias a la publicación de *Ten Lectures on Wavelets* [18] por Ingrid Daubechies en 1992 que se pudieron formalizar estos conceptos.

Dicha publicación permitió revolucionar el campo de la compresión de datos, dando pie a que los avances que ya se habían hecho con transformadas de Fourier fueran aplicables en un abanico más amplio de ramas de conocimiento.

En las siguientes secciones, se presentan ejemplos ilustrativos en los que se hace uso de bases de Fourier y *Wavelets* para conseguir representaciones muy dispersas en una y dos dimensiones. Puesto que estudiar en profundidad estos resultados nos alejaría del propósito del trabajo, se proporciona [5] como recurso para consultar estos resultados con más detalle.

3.2. Compresión de señales

3.2.1. Compresión de señales 1-D

En el caso unidimensional, podemos beneficiarnos del fenómeno de dispersión para cosas tan importantes como conseguir eliminar el ruido habitual que se genera al captar una señal y conservar tan sólo la señal original que se quiere transmitir, proceso que se conoce como **filtrado** de la señal. En el siguiente ejemplo, haremos uso del algoritmo **FFT** con este propósito.

Aunque no entremos en detalles de su formulación, es importante mencionar que el algoritmo *Fast Fourier Transform* (FFT) fue desarrollado por James W. Cooley y John W. Tukey en 1965 [19] partiendo de un algoritmo anterior, el *Discrete Fourier Transform* (DFT) que esencialmente discretizaba uniformemente una señal continua en el dominio de la frecuencia, escogiendo como base una combinación de funciones sinusoidales.

Pese a que DFT resultaba eficaz para aproximación numérica y computación, al tomar una muestra de tamaño n , requiere de un número de operaciones de $O(n^2)$, lo que supone un coste computacional inafrontable para valores de n muy altos.

Mediante un juego de simetrías en las transformadas de Fourier, Cooley y Tukey fueron capaces de resolver una DFT de tamaño n mediante la computación de DFTs de dimensiones menores, reduciendo el número de operaciones a $O(n \log n)$ y dando lugar al FFT [6].²

²Aunque pueda parecer un pequeño cambio, computacionalmente hablando aporta una diferencia abismal, puesto que para valores de n del orden de 10^6 con los que habitualmente trabajaremos, el valor de $\log(n)$ crece a un ritmo mucho más lento, permitiendo al algoritmo seguir un crecimiento prácticamente lineal. Este hecho revolucionó el campo de compresión de datos, y sobre él se sustentan muchos de los formatos de uso cotidiano como MP3 o JPEG.

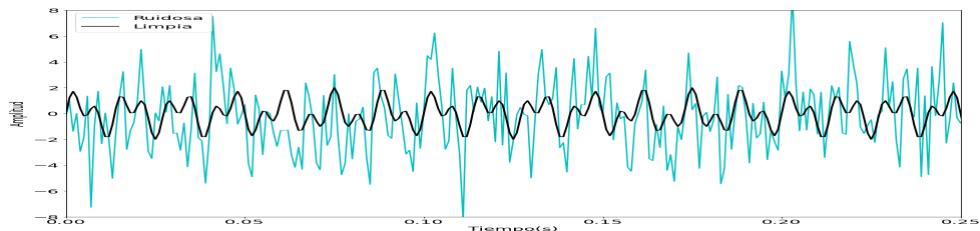
Ejemplo 3.2.1. Proporcionamos ahora un ejemplo sencillo de cómo el algoritmo FFT puede ayudar a filtrar el ruido de una señal implementado en Python: partiendo de la señal original (a la que hemos introducido previamente ruido aleatorio), podemos apoyarnos en FFT para "limpiar" la señal, conservando solamente la señal original tras invertir el cambio al espectro del tiempo.

Para no complicar innecesariamente el ejemplo y que sea ilustrativo, consideraremos la señal continua en función del tiempo:

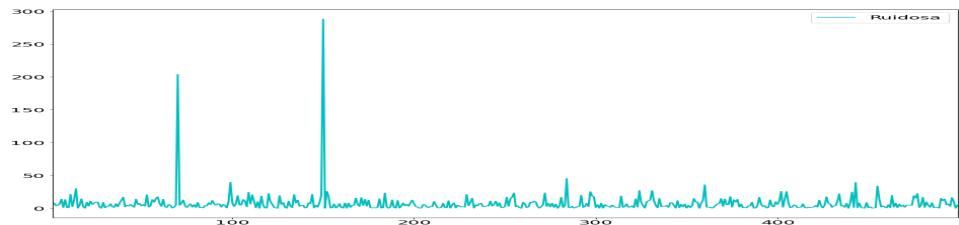
$$x(t) = \sin(2\pi 70t) + \sin(2\pi 150t) \quad (3.1)$$

compuesta por dos frecuencias: una de 70 Hz y otra de 150 Hz, a la que añadimos una gran cantidad de ruido blanco³ como se puede observar en la Figura 3.1a:

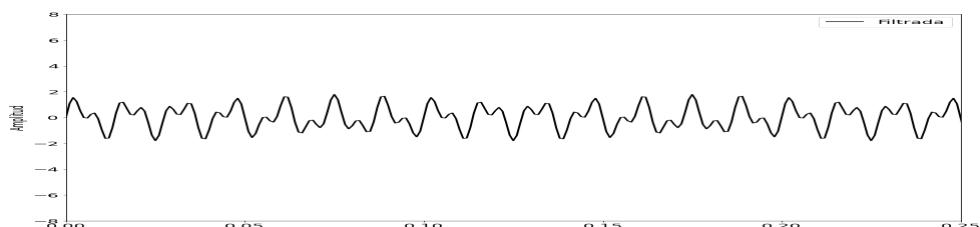
Figura 3.1: Eliminación de ruido de una señal unidimensional mediante el algoritmo FFT



(a) Representación en el dominio del tiempo de la señal con y sin ruido.



(b) Señal 2-dispersa con muchos coeficientes cercanos a cero debido al ruido blanco.



(c) Recuperación de la señal original tras el filtrado.

Haciendo uso de la librería *numpy*, podemos computar el algoritmo FFT y representar nuestra señal en el dominio de la frecuencia, mostrando de este modo que se trata de una señal 2-dispersa al comprobar la existencia de dos picos, correspondientes a las dos frecuencias de la señal original (ver Figura 3.1b).

Procediendo como se explicó en la Observación 3.1.2, consideramos los valores cercanos a cero indistintamente como nulos, obteniendo así la señal limpia compuesta únicamente por los dos picos. A

³El ruido blanco es una señal aleatoria que se caracteriza por el hecho de que sus valores de señal en dos tiempos diferentes no guardan correlación estadística.

esta señal le aplicamos la inversa del algoritmo FFT para volver al dominio del tiempo, recuperando así la señal $x(t)$ filtrada (Figura 3.1c).

3.2.2. Compresión de señales 2-D

Entenderemos por señales 2-D matrices $x \in \mathbb{R}^{n \times m}$ que representen una imagen de $n \times m$ píxeles: a cada píxel vendrán asociados unos ciertos valores correspondientes a los distintos canales de color pertenecientes al modelo cromático en el que se esté representando la imagen. Aunque el más habitual es el modelo *Red, Green & Blue (RGB)*, que consta de 3 canales, nosotros trabajaremos con imágenes en escala de grises para disponer de toda la información en un único canal con el objetivo de simplificar los cálculos. No obstante, es importante mencionar que en caso de utilizar el modelo RGB, habría que trabajar con una dimensión adicional en la que habría 3 componentes.

De igual modo que en 1-D, el algoritmo FFT puede implementarse en 2-D: dada una señal x , se ejecuta primero el algoritmo unidimensional sobre todas las filas de la matriz y después sobre todas las columnas de la matriz resultante.

No obstante, en los últimos años los avances en la teoría de *Wavelets* han eclipsado al FFT: mediante la extensión de los conceptos de análisis de Fourier a bases ortonormales más genéricas, se pueden descomponer señales más complejas que aparecen en procesos multi-escala en los que el contenido en el espectro de frecuencias no es estacionario, lo que permite simultáneamente caracterizar el contenido en frecuencias y su evolución en el tiempo. Estos pueden aparecer en un abanico muy amplio de ramas de conocimiento, en particular en geosísmica, además de ser la base de servicios en *streaming* como Netflix, por ejemplo [7].

Ejemplo 3.2.2. Presentamos a continuación un ejemplo en el que una imagen es comprimida y descomprimida usando el algoritmo *Discrete Wavelet Transform (DWT)*, y mas concretamente la clase de ondículas conocida como **Wavelets de Daubechies** [20], caracterizados por formar un sistema ortonormal de $L^2(\mathbb{R})$, además de poseer todos sus elementos un soporte compacto.⁴

Para su implementación haremos uso del paquete *PyWavelets* [36] de Python, lo que nos permitirá en primer lugar construir una intuición de la descomposición que hace de la Figura 3.2a el algoritmo DWT: encontramos en la esquina superior izquierda de la Figura 3.2b una versión de la imagen de poca resolución a la que van siendo añadidos detalles conforme el porcentaje de coeficientes conservados aumenta (lo que implica mayor peso del archivo comprimido)⁵. También, el mismo algoritmo nos permite apreciar la gran dispersión de la señal en el dominio de *Wavelets* de Daubechies, la cual se hace evidente al reconstruir la imagen original partiendo de distintos porcentajes de los coeficientes de mayor valor que se va reduciendo (ver Figura 3.3).

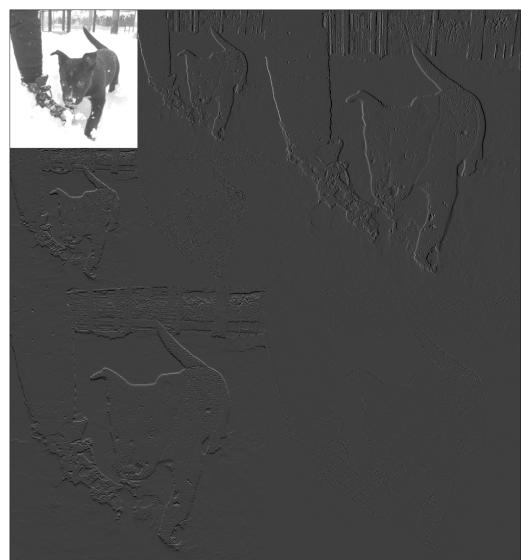
⁴Este tipo de ondículas es, junto con los *Wavelets* de Haar, los que más trataremos a lo largo del trabajo.

⁵Intuitivamente, podemos relacionar esto con la calidad de imagen que recibimos en nuestros ordenadores mientras hacemos uso de algún servicio en *streaming*: cuanto mayor es el ancho de banda, mayor es la calidad, pero aún en condiciones muy austeras, el algoritmo permite esbozar los trazos fundamentales de la imagen; la esquina superior izquierda de la que hablábamos.

Figura 3.2: Representación de una señal 2-D en el dominio de *Wavelets* de Daubechies.



(a) Imagen original.



(b) Dominio de *Wavelets* de Daubechies.

Figura 3.3: Compresión y descompresión de una señal 2-D conservando distintos porcentajes de los coeficientes de **mayor valor** en el dominio de *Wavelets de Daubechies*. Nótese que la conservación del 5 % de los coeficientes proporciona una reconstrucción a simple vista perfecta, y no es hasta que reducimos hasta el 1% que los cambios bruscos de color y textura empiezan a estar más difusos en las fronteras.



(a) Conservando el 10 % de los coeficientes.



(b) Conservando el 5 % de los coeficientes.

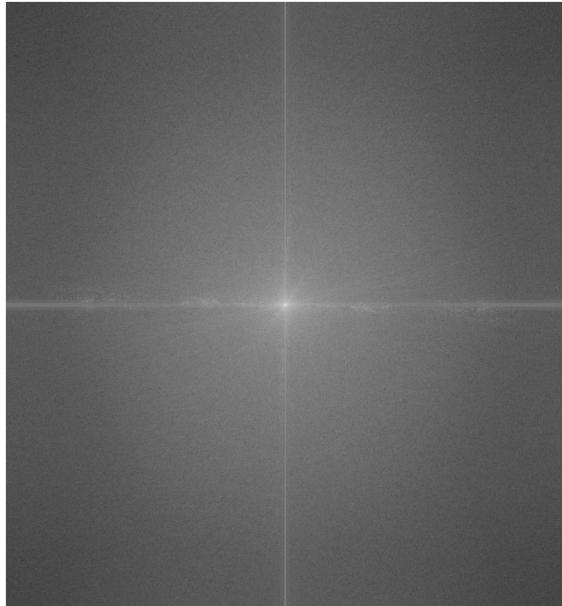


(c) Conservando el 1 % de los coeficientes.



(d) Conservando el 0,5 % de los coeficientes.

Figura 3.4: Adicionalmente, incluimos la representación de la misma imagen en el dominio de Fourier (aplicando el algoritmo FFT 2-dimensional) y su reconstrucción a partir del descarte de los coeficientes de menor valor en este dominio con el objetivo de evidenciar la mejoría que proporcionó el descubrimiento de los *Wavelets*.



(a) Representación en el dominio de Fourier.



(b) Conservando el 5 % de los coeficientes.



(c) Conservando el 1 % de los coeficientes.



(d) Conservando el 0,5 % de los coeficientes.

Capítulo 4

Compressed Sensing

Aprovechándose de los avances en tratamiento de señales realizados en los últimos años, la técnica *Compressed Sensing*, también conocida como *Compressed Sampling* o por sus siglas CS, nace de afrontar el paradigma de la compresión desde otro enfoque: en vez de captar cada señal al completo para proceder después a su compresión, en la que gran parte de los coeficientes serán descartados debido a la gran dispersión que presentan en un *frame* adecuado, una cierta señal x originariamente de dimensión n que sea K -dispersa (respecto de cierto *frame* Ψ) puede ser reconstruida **con gran probabilidad**¹ habiendo captado previamente solo una **pequeña muestra** de la señal x , en adelante $y \in \mathbb{R}^m$, cumpliéndose que $K < m \ll n$.

Aunque matemáticamente la idea tras la técnica no es muy compleja, determinar el número m de coeficientes que componen y que aseguran una reconstrucción exitosa necesita de varias premisas que estudiaremos en este capítulo, siendo la elección de la matriz de captación el concepto fundamental.

4.1. Matriz de captación

Definición 4.1.1 (Matriz de captación). Sea $x \in \mathbb{R}^n$ una señal K -dispersa en el *frame* Ψ , entonces bajo las premisas de *CS*, podremos encontrar $y \in \mathbb{R}^m$ de manera que

$$y = \Phi x, \quad (4.1)$$

donde $K < m \ll n$ y la matriz $\Phi \in \mathbb{R}^{m \times n}$ recibirá el nombre de **matriz de captación**, cuya elección será fundamental como veremos en las secciones venideras.

Observación 4.1.2. Una interpretación de la matriz de captación muy intuitiva es pensar en cada una de las filas que la componen (en adelante ϕ_i) como una captación independiente, lo que nos conduce a la pregunta de cómo determinar el menor número m de captaciones que permiten la reconstrucción de una cierta señal.

Dado que nuestra señal x es K -dispersa respecto de Ψ , una posibilidad para proceder a su reconstrucción sería mediante el cálculo de sus coordenadas respecto de Ψ , que denotaremos por $\alpha \in \mathbb{R}^n$, atendiendo a la ecuación

$$y = \Phi \Psi \alpha \quad (4.2)$$

¹La reconstrucción de la señal mediante la técnica *CS* no está garantizada, por lo que el papel que juega la teoría de probabilidad es fundamental. No obstante, bajo las premisas que veremos a lo largo del capítulo, la probabilidad de reconstrucción exitosa es muy alta para problemas de dimensiones moderadas.

Puesto que de antemano no conocemos la posición de las K componentes del vector α no nulas, es primordial que la matriz de captación sea extrapolable a cualquier problema de la misma dimensión, por lo que la composición de la matriz de captación debe seguir un procedimiento general, y no ser particular al problema que estamos tratando de resolver.

Nótese que el sistema (4.2) es compatible, pero en principio indeterminado por lo que, en general, existirán una infinidad de vectores α que cumplan nuestros requisitos.

No obstante, puesto que a priori conocemos la dispersión presente en nuestra señal, podemos reformular el problema que acabamos de describir como la determinación de $\hat{\alpha}$, donde:

$$\hat{\alpha} = \min \|\alpha\|_0, \text{ sujeto a } y = \Phi\Psi\alpha \quad (4.3)$$

donde $\|\cdot\|_0$ es la aplicación que se introdujo en la Observación 2.1.12.

Este es un problema **no convexo**, cuya solución solo podríamos determinar a base de probar todos los posibles vectores K-dispersos en \mathbb{R}^n , por lo que su resolución queda descartada para dimensiones moderadamente grandes de n y K .

Hasta mediados de la década de los 2000, el problema de encontrar la solución de menor dispersión era considerado un problema NP-hard² y por lo tanto, no era factible encontrar una solución con un coste computacional abordable. Gracias al trabajo de Emmanuel Candès y Terrance Tao [22] entre otros, se consiguió determinar las premisas bajo las cuales es posible la reconstrucción total de la señal usando algoritmos convexos.

Llegamos de este modo a la primera afirmación realmente novedosa de la técnica, referente a la dimensión de captación [8]:

Proposición 4.1.3. Dada una cierta señal $x \in \mathbb{R}^n$ que sea K-dispersa respecto del *frame* Ψ , habiendo realizado un número de captaciones m tal que

$$m \geq C \cdot K \cdot \log(n/K) \quad (4.4)$$

donde C es una cierta constante positiva cuyo valor dependerá de la **incoherencia** (que estudiaremos en la Subsección 4.2.1) existente entre el par de matrices Φ, Ψ , la solución a nuestro problema (4.3) puede ser determinada mediante la resolución de un problema convexo de minimización ℓ_1 [23] de la forma:

$$\hat{\alpha} = \min \|\alpha\|_1, \text{ sujeto a } y = \Phi\Psi\alpha \quad (4.5)$$

que mediante la implementación de varios algoritmos puede ser resuelto.

Remark. En el Apéndice A se proporciona un método para la reformulación de problemas de minimización ℓ_1 como problemas de programación lineal.

Para construir una intuición de cómo la minimización de la norma ℓ_1 tiende a llegar a soluciones de gran dispersión, consideremos el siguiente ejemplo dos dimensional:

Ejemplo 4.1.4. Sea $x \in \mathbb{R}^2$ una señal que quiere ser aproximada con un elemento del espacio afín unidimensional A , que en este caso es una recta. El elemento que minimiza el error cometido habiendo inducido una norma ℓ_p será:

$$\hat{x} \in A : \min \|x - \hat{x}\|_p$$

es decir, el elemento de A que proporciona la menor distancia a x . Para minimizar dicha norma, construimos una esfera \mathbb{S} (con la norma correspondiente) centrada en x hasta intersecar con A , llegando así al valor de \hat{x} .

²El conjunto de problemas de decisión computacional que son como mínimo tan difíciles como un problema NP (*nondeterministic polynomial time*) [21].

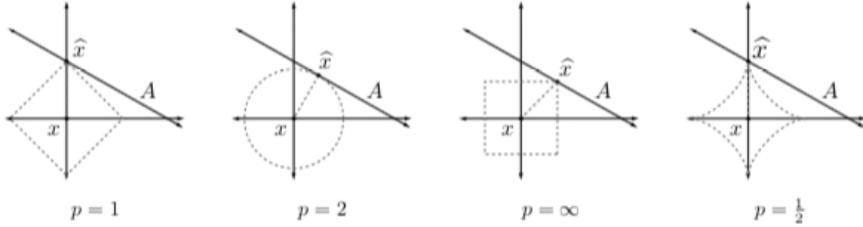


Figura 4.1: Minimización de la distancia entre x y A con distintas (cuasi)-normas ℓ_p inducidas.

En la Figura 4.1 podemos observar cómo los valores de $p = 2$ y $p = \infty$ tienden a distribuir el error cometido entre ambas componentes, mientras que valores de $p \leq 1$ concentran el error en una única componente, favoreciendo la dispersión. Este concepto se puede generalizar para dimensiones de n arbitrarias y juega un papel fundamental en el desarrollo de la técnica [9].

En las secciones venideras veremos varios ejemplos en los que afrontaremos la reconstrucción mediante la implementación de algoritmos ℓ_1 como el *Orthant-wise limited memory quasi-Newton (OWL-QN)* [24], que es una variante del *Limited-memory BFGS (LBFGS)* [25] específicamente diseñado para optimizar problemas explotando la dispersión inherente de la señal (en la Subsección 4.4.2 se proporciona un ejemplo de su implementación).

Observación 4.1.5. Además de la resolución mediante minimización ℓ_1 , que requiere de un coste computacional muy elevado, los conocidos como algoritmos voraces o *Greedy*³ se postulan en ocasiones como una opción más viable. Estos abordan el problema directamente formulado con la norma ℓ_0 y en la base de captación original:

$$\min \|y - \Phi x\|_2, \text{ sujeto a } \|x\|_0 \leq K$$

donde se intenta hacer una elección de la matriz de captación mediante iteraciones que minimicen el error euclídeo (bajo situaciones de ruido), aunque no son tan fiables en cuanto a encontrar la solución óptima. Entre los que mejores resultados han conseguido obtener, destacan *Orthogonal Matching Pursuit* [26] y *CoSaMP* [27].

Es importante resaltar que el papel de la teoría de probabilidad en la técnica es **fundamental**, y por ello se han determinado ciertas condiciones de compatibilidad que deben de existir entre las matrices Φ y Ψ para que la probabilidad de que la solución al problema (4.5) coincida con la del problema original (4.3) sea suficientemente alta como para proporcionar una reconstrucción exitosa.

³Un algoritmo *Greedy* es una estrategia de búsqueda por la cual se sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima.

4.2. Condiciones de compatibilidad

Además de una matriz de captación Φ convenientemente elegida, debemos de exigir ciertas condiciones de compatibilidad entre las matrices Ψ y Φ que permitan asegurar que la dispersión de la señal se preserva al representarla respecto de ambas referencias. Para poder cuantificar este concepto, se definió el coeficiente de **coherencia** [8].

4.2.1. Coherencia

Definición 4.2.1. La **coherencia** μ entre el par de matrices Φ, Ψ viene determinada por la siguiente expresión:

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq i, j \leq n} |\langle \phi_i, \psi_j \rangle| \quad (4.6)$$

, siendo ϕ_i la i -ésima fila de Φ y ψ_j la j -ésima columna de Ψ .

El valor de μ pertenecerá al intervalo $[1, \sqrt{n}]$ y nos indicará la correlación existente entre coeficientes de ambas familias.

Definición 4.2.2. Las matrices Φ, Ψ son **poco coherentes** cuando el valor de $\mu(\Phi, \Psi)$ es muy próximo a 1. Por el contrario, las matrices Φ, Ψ son muy coherentes cuando el valor de $\mu(\Phi, \Psi)$ es muy próximo a \sqrt{n} .

La técnica *CS* está enfocada a pares de matrices **poco coherentes** lo que permitirá que el valor de la constante C de la Ecuación 4.4 sea muy cercano a 1, y por tanto que el número de captaciones necesarias para obtener una reconstrucción exitosa sea del orden de $K \log(n/K)$.

Observación 4.2.3. Intuitivamente puede ayudar pensar en el caso en que Ψ es una base ortonormal, ya que debido a que sus columnas son ortogonales, si el valor de μ es cercano a \sqrt{n} (situación derivada de que existan valores de $|\langle \phi_i, \psi_j \rangle|$ cercanos a 1), muchos vectores α candidatos a ser el de menor dispersión que satisface las hipótesis de (4.5) son descartados al contener la matriz producto $\Phi\Psi$ (que en adelante denotaremos por la letra Θ) filas en las que muchos elementos se anulan. Esto se debe a que, como las columnas de Ψ (ψ_j) son ortogonales, en caso de que se cumpla que $|\langle \phi_i, \psi_j \rangle| = 1$, entonces para todos los valores de $k \neq j$ se tiene que $|\langle \phi_i, \psi_k \rangle| = 0$.

La Figura 4.2 emula la situación que acaba de ser descrita:

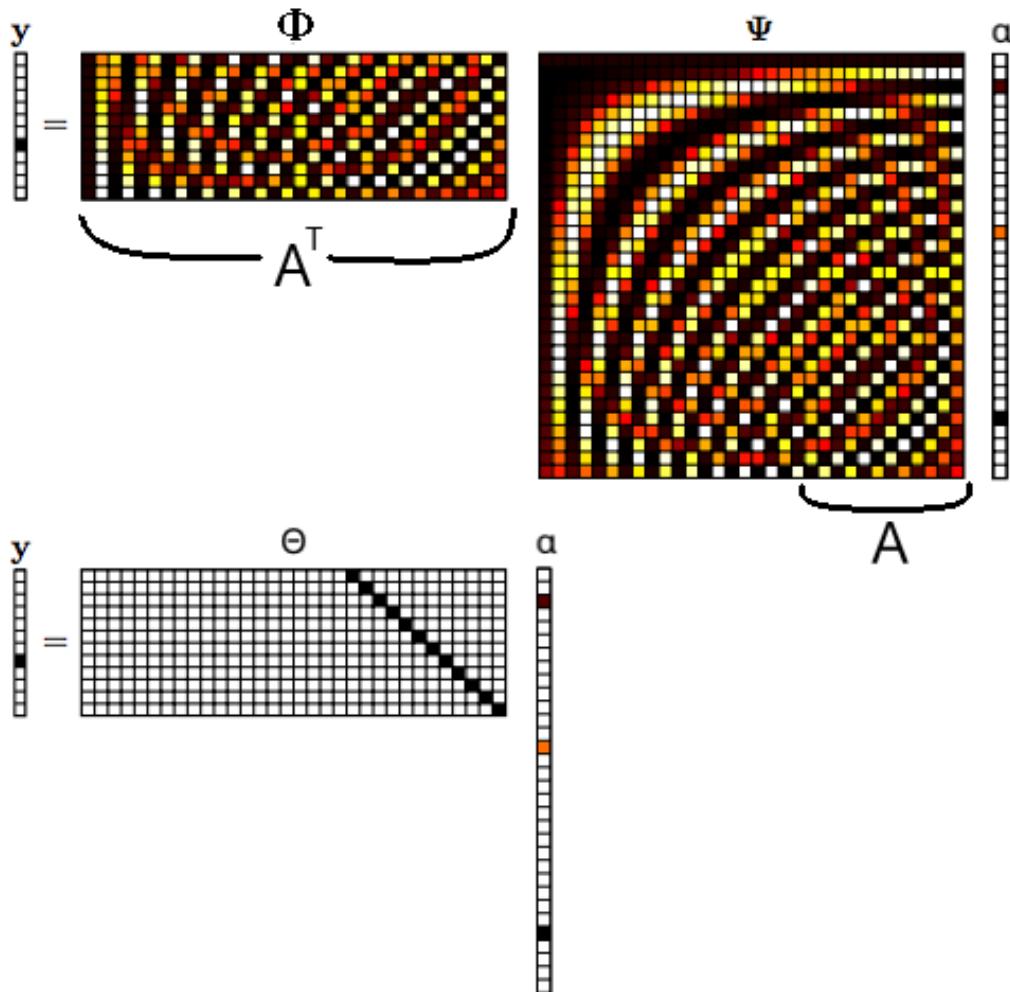


Figura 4.2: Situación derivada de un par de matrices coherente: mediante el uso de un código de colores en el que cada tonalidad representa un número diferente, procedemos a llamar A a la matriz formada por las últimas p columnas de Ψ . Si elegimos Φ como la matriz transpuesta de A , entonces Θ queda compuesta por dos submatrices de p filas: las $(n - p)$ primeras columnas formadas por ceros y la matriz identidad de dimensión p en las últimas columnas. Por lo tanto, cualquier candidato α cuyas ultimas p componentes sean nulas quedaría descartado aún siendo el de mayor dispersión.

4.2.2. Restricted Isometry Property

Cuando el par de matrices Φ, Ψ con el que trabajamos es suficientemente incoherente, Θ satisface la propiedad de restricción isométrica, conocida como **RIP** (*Restricted Isometry Property*), que fue introducida por primera vez en [28], y que establece la existencia de una constante $\delta_K \in (0, 1)$ tal que para todos los vectores α K-dispersos se cumple que:

$$(1 - \delta_K) \|\alpha\|_2^2 \leq \|\Theta\alpha\|_2^2 \leq (1 + \delta_K) \|\alpha\|_2^2 \quad (4.7)$$

es decir, asegura que la matriz Θ actúa **casi** como una isometría⁴ para los vectores K-dispersos. En la práctica, el cálculo directo de δ_K es altamente complicado, por lo que la elección de una Φ particular para tratar cada caso no es un camino factible. Por este motivo, la elección de Φ debe de tener siempre un **grado de aleatoriedad**, lo que permitirá sacar conclusiones sobre una captación de dimensión m genérica, permitiendo de este modo acotar el valor de δ_K para una cierta familia de matrices de representación.

Generalmente, cuanto mayor es la dimensión de la matriz de captación, menor es el valor de δ_K , lo que permite que Θ actúe como una isometría para vectores dispersos. Bajo esta premisa, podemos asegurar que la geometría (y por tanto la dispersión) de la señal a comprimir se conserva al hacer una reducción de la dimensión del problema estudiado, hecho que también se concluye del Lema de Johnson-Lindenstrauss: la incoherencia en el contexto de *frames* equivale (aproximadamente) a la ortogonalidad en el contexto de bases [29].

Por lo tanto, cuando trabajamos con un par de matrices cuyo coeficiente de coherencia es cercano a 1 y se han realizado suficientes captaciones, es posible determinar detalladamente las K componentes no nulas del vector n -dimensional α , lo que nos permite determinar los valores de δ_K para los que la exactitud de la reconstrucción está garantizada.

Observación 4.2.4. Realmente, la condición RIP no deja de ser una particularización de la Ecuación 2.9 en la que exigimos que nuestro *frame* se comporte de manera similar a un *Frame de Parseval*. En el caso idílico, el valor de $\delta_K = 0$, y por lo tanto estaríamos trabajando con un *Frame de Parseval*, por lo que la preservación de la geometría estaría garantizada.

Para más detalles sobre la condición RIP, se recomienda consultar [30].

4.3. Matrices de captación universales

Uno de los avances mas importantes a la hora de ampliar el abanico de campos en el que la técnica es aplicable, fue la determinación de ciertas matrices de captación que forman un par poco coherente con una gran variedad de bases de representación, conocidas como **matrices de captación universales**. Comenzamos por presentar un resultado relacionado con bases de Fourier y *Wavelets*, puesto que como se estudió en el Capítulo 3 una gran variedad de señales naturales presentan una representación de muy dispersa en sus respectivos dominios.

Proposición 4.3.1. La matriz de captación Φ compuesta por m filas ϕ_i seleccionadas de manera aleatoria de la matriz identidad I_n , presenta **máxima incoherencia** junto a Ψ compuestas por bases de Fourier o *Wavelets*.

Demostración. Puede encontrarse en [8]. □

⁴Una isometría es una aplicación entre dos espacios métricos que conserva la distancia entre los puntos. En nuestro caso, esto ayudará a preservar la estructura del producto escalar en el espacio de señales, y por tanto la norma euclídea de los vectores k-dispersos.

Abordando ahora casos en los que Ψ no es necesariamente una de las anteriores, en su tesis doctoral [10] Mark Davenport hizo grandes avances en la determinación de matrices de captación universales. En particular, trabajó con Φ formadas mediante la elección reiterada de sucesos independientes e idénticamente distribuidos siguiendo distintas distribuciones notables llegando a resultados muy interesantes, de entre las que destacamos la siguiente

Proposición 4.3.2. Si fijado un δ_K , formamos nuestra matriz de captación siguiendo alguna de las siguientes distribuciones notables:

- Bernoulli
- Gaussiana

y el número de captaciones m es tal que se cumple la expresión:

$$p \geq K_1 m \log\left(\frac{n}{m}\right)$$

entonces la condición RIP se satisface con probabilidad $1 - 2e^{-K_2 p}$, siendo K_1 arbitraria y

$$K_2 = \frac{\delta_m^2(1 - \log(2))}{4} - \frac{\log(42e/\delta_m)}{K_1}$$

Demostración. Ver la tesis doctoral de Mark Davenport [10]. □

4.4. Primeras aplicaciones de la técnica

Presentamos a modo de introducción a la técnica ejemplos en una y dos dimensiones, con el objetivo de clarificar los conceptos antes de dar paso a los casos reales en el campo de geosísmica.

4.4.1. Compressed Sensing para señales 1-D

Presentamos dos ejemplos para mostrar tanto la reconstrucción mediante minimización de norma ℓ_1 como mediante un algoritmo Greedy que consigue una reconstrucción exitosa aún con un pequeño porcentaje de captación:

Ejemplo 4.4.1. A diferencia de como hicimos en el Capítulo 3, partiremos de una señal continua (en función del tiempo) libre de ruido, esta vez compuesta por la suma de 4 funciones sinusoidales:

$$x(t) = \sin(2\pi 70t) + \frac{3}{4} \sin(2\pi 100t) + \frac{3}{5} \sin(2\pi 150t) + \frac{3}{8} \sin(2\pi 200t) \quad (4.8)$$

correspondientes a las frecuencias 70 Hz, 100 Hz, 150 Hz y 200 Hz respectivamente.

Por lo tanto, su representación en el espectro de la frecuencia presentará 4 picos, y diremos que nuestro vector $x \in \mathbb{R}^{1000}$ (puesto que hemos tomado $n = 1000$ valores discretos de la señal), correspondiente a la discretización de $x(t)$, es una señal 4-dispersa, como podemos observar en la Figura 4.3.

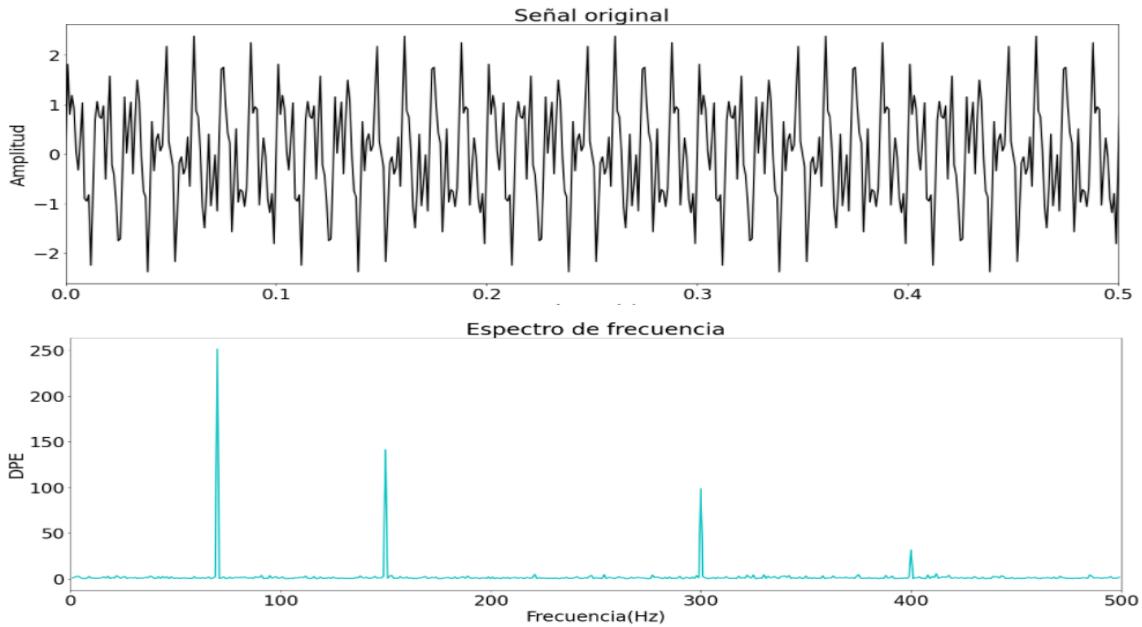


Figura 4.3: Representación de una señal 4-dispersa en el dominio de frecuencias.

Siguiendo lo visto en la Proposición 4.3.1, optamos por la elección de una Φ compuesta por m filas aleatorias de la matriz identidad I_n , lo que nos aportará máxima respecto a nuestra Ψ , compuesta por una matriz de cambio de base del dominio de frecuencias al temporal.

Como se vio en la Subsección 4.2.1, la elección de una dimensión de captación $m = 30$ debe de proporcionar una reconstrucción exitosa, puesto que en ese caso, $m \geq 4 \cdot \log(1000/4) \approx 23$.

Una vez conocemos los valores de las dimensiones a ejecutar, procedemos a la captación de los $y = 30$ valores aleatorios en nuestra señal a partir de los cuales procederemos a la reconstrucción (ver Figura 4.4).

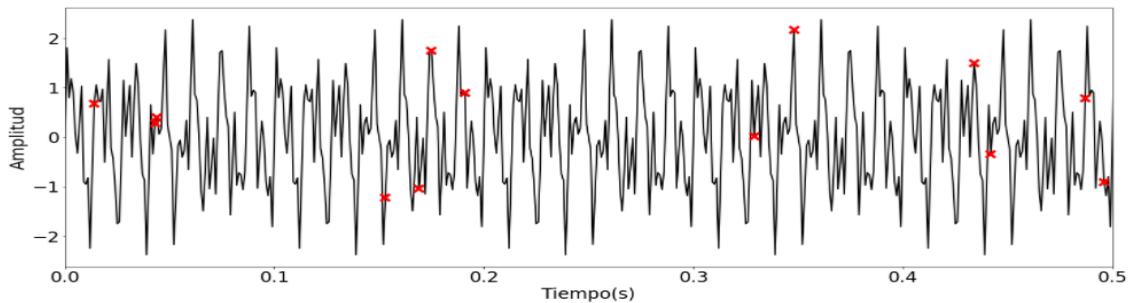


Figura 4.4: Captaciones aleatorias sobre una señal 1-D.

Apoyándonos en la librería CVXPY [37],[38] para la optimización ℓ_1 , en la Figura 4.5 resolvemos el problema de minimización (4.5), cuya duración de ejecución aproximada son 0.25 segundos. Pese a que la reconstrucción no puede considerarse perfecta, es satisfactoria teniendo en cuenta que se había captado previamente tan solo un 3 % de la señal total. Adicionalmente, se presenta la reconstrucción mediante minimización de la norma euclídea (ℓ_2), y se aprecia a simple vista que no es capaz de proporcionar una reconstrucción aproximada de la señal.

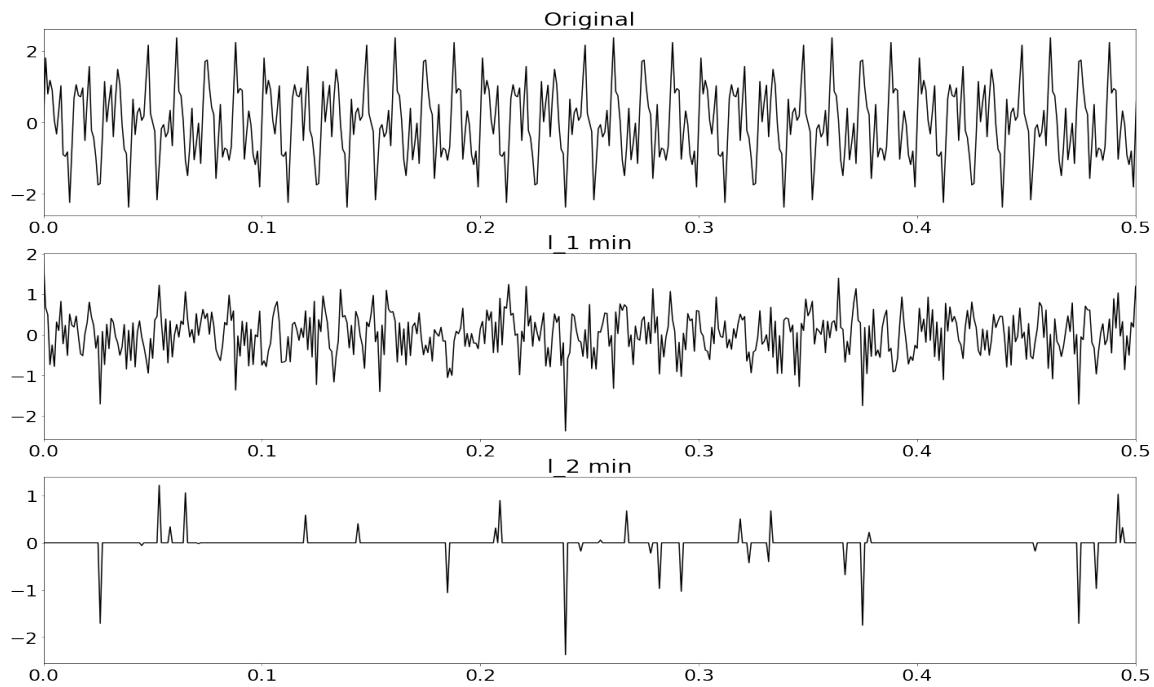


Figura 4.5: Reconstrucción de una señal 1-D mediante minimización de norma ℓ_1 y ℓ_2 .

Ejemplo 4.4.2. Presentamos ahora un ejemplo en el que hacemos uso del algoritmo *CoSaMP* [27], que es un algoritmo voraz (ver Observación 4.1.5), para la reconstrucción de otra señal unidimensional con el objetivo de comparar con la reconstrucción mediante minimización ℓ_1 .

En esta ocasión, optamos por una señal formada por la suma de cuatro funciones sinusoidales de mayor variedad en cuanto a las frecuencias presentes para comprobar si la reconstrucción es exitosa en condiciones de mayor variación:

$$x(t) = \cos(2\pi 97t) + \frac{3}{4}\cos(2\pi 350t) + \frac{1}{2}\cos(2\pi 600t) + \frac{2}{5}\cos(2\pi 877t) \quad (4.9)$$

y en la Figura 4.6 comprobamos que consigue una reconstrucción perfecta al mismo tiempo que ofrecer un coste computacional muy moderado al tratarse de un algoritmo cuyas iteraciones están compuestas solamente por multiplicaciones matriz-vector reiteradas, requiriendo de un tiempo de reconstrucción del orden de $O(n \log^2 n)$, donde n es la longitud de la señal.

Esto le permite ofrecer, bajo las mismas premisas que en el Ejemplo 4.4.1 ($n = 1000, m = 30$), reconstrucciones de señales en n un tiempo aproximado de 0,03 segundos.

Además de la señal original y reconstruida en el dominio temporal, aportamos en la Figura 4.7 la representación en el dominio de la frecuencia de ambas señales para comprobar la exactitud de la reconstrucción.

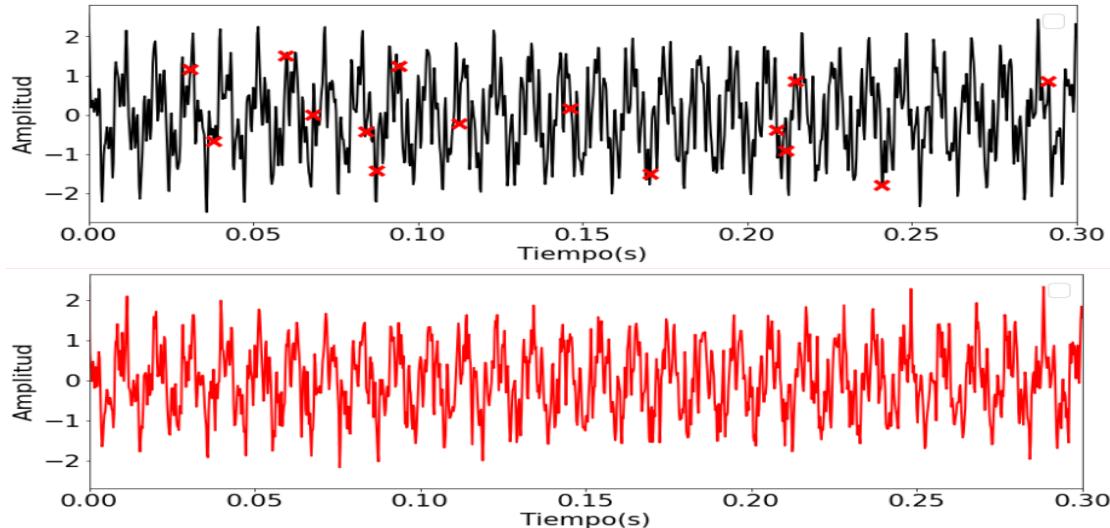


Figura 4.6: Reconstrucción de señal 1-D mediante el algoritmo CoSaMP.

Observación 4.4.3. Antes de dar paso a tratar casos dos dimensionales, es conveniente enfatizar la mejora que aporta la técnica: previamente al desarrollo de *CS*, la única forma de recuperar una señal como la que acabamos de mostrar sería siguiendo la tasa de Nyquist: puesto que la frecuencia más alta es $F = 877$ Hz, requeriríamos de una frecuencia de muestreo superior al doble de F , es decir, deberíamos de realizar más de 1700 captaciones por segundo, mientras que siguiendo la técnica hemos conseguido realizar la reconstrucción con tan solo 30 captaciones en el mismo intervalo de tiempo.

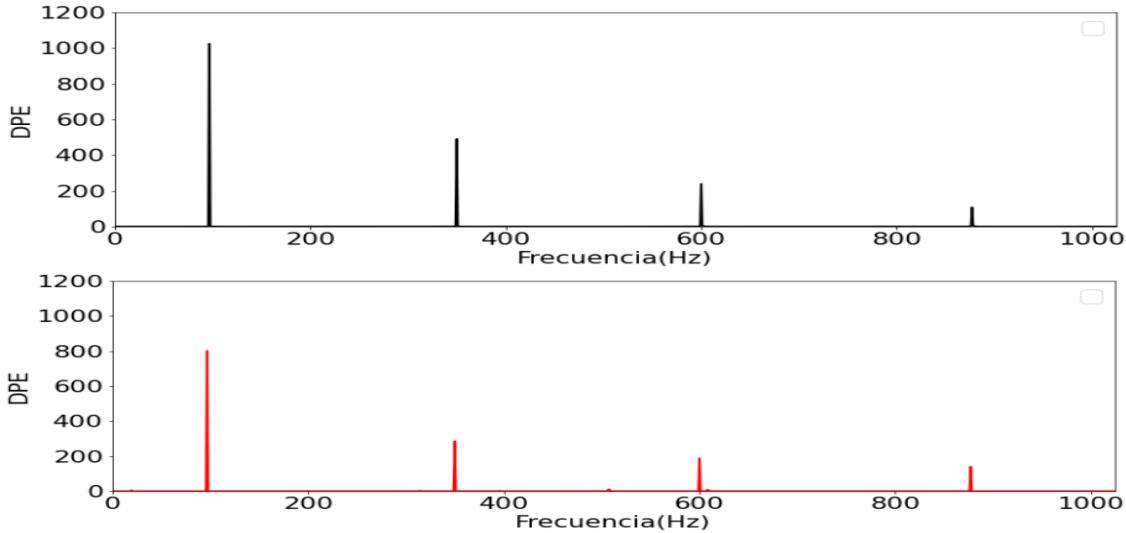


Figura 4.7: Espectro de frecuencias de la señal original y reconstruida mediante CoSaMP.

4.4.2. Compressed Sensing para señales 2-D

Procedemos a reconstruir en la Figura 4.8 la imagen con la que hemos trabajado en secciones anteriores a partir de una muestra aleatoria del 10 % de la señal mediante la implementación del algoritmo *Orthant-Wise Limited-memory Quasi-Newton* (OWL-QN), disponible en la librería *PyLBFGS*, escrita por Robert Taylor⁵.

Aunque la documentación completa se puede encontrar en [24], es importante resaltar que para su implementación, es necesario reformular primero el problema de optimización a minimizar para que sea de la forma:

$$f(x) = g(x) + C \|x\|_1 \quad (4.10)$$

, donde $C \in \mathbb{R}$ y $g(x)$ debe de ser necesariamente una función convexa y diferenciable.

En nuestro caso, la función $g(x)$ será el error euclídeo cometido al reconstruir tan solo la parte de la señal correspondiente a las captaciones de las que se tiene constancia, es decir, la muestra y .

Por tanto, (4.10) se puede reformular para ser consistentes con la notación que veníamos utilizando como:

$$f(\alpha) = \|\Theta\alpha - y\|_2^2 + \|\alpha\|_1 \quad (4.11)$$

Adicionalmente, el algoritmo requiere de la expresión del gradiente de $g(x)$, que en nuestro caso viene determinada por:

$$\nabla g(\alpha) = 2\Theta^T(\Theta\alpha - y)$$

Se opta por la elección de la transformada *Discrete Cosine Transform*(DCT) como cambio de referencia, la cual se caracteriza por estar compuesta por series de Fourier que toman valores en el dominio de los números reales, lo que permite enfatizar los componentes de baja frecuencia, favoreciendo de este modo la preservación de la señal aún cuando el porcentaje de compresión es alto [31].

⁵También está disponible en <https://www.microsoft.com/en-in/download/details.aspx?id=52452> distribuida por Microsoft.

Por lo tanto, la matriz Θ se corresponde con la inversa de la transformada DCT (para pasar del dominio de frecuencia al temporal) mientras que $\Theta^T = \Theta^{-1}$ es la transformada DCT, puesto que trabajamos con una base ortonormal.

A partir de una muestra aleatoria del 10 % de los coeficientes, el algoritmo consigue reconstruir con gran exactitud la imagen original en un tiempo de ejecución aproximado de 54 segundos:

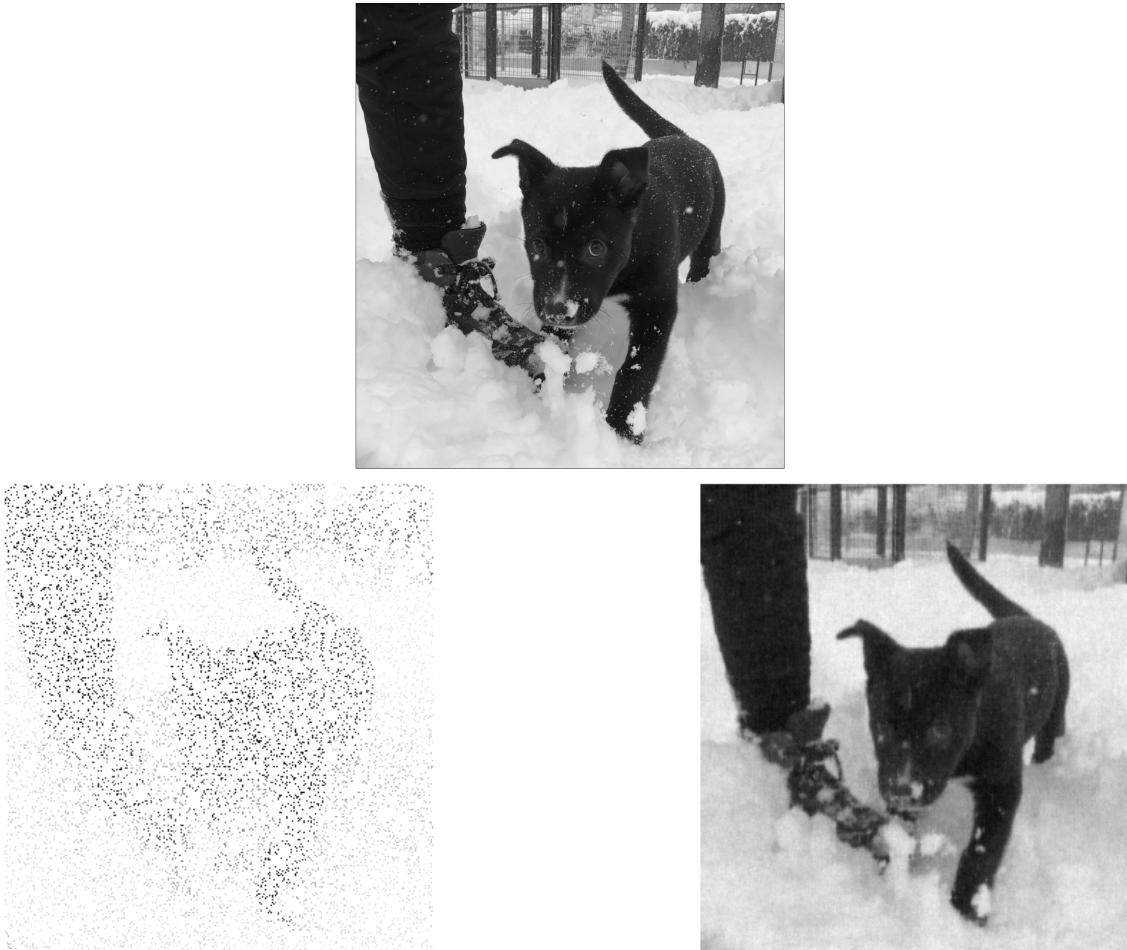


Figura 4.8: Reconstrucción mediante el algoritmo OWL-QN de una imagen partiendo de una muestra aleatoria del 10 %.

Capítulo 5

Compressed Sensing para adquisición sísmica

En este capítulo aplicaremos los conceptos que se han expuesto en el capítulo anterior a situaciones pertenecientes al campo de geofísica. La situación inicial del problema a estudiar es la siguiente: al afrontar una posible extracción de petróleo, se procede a la colocación de **geófonos** en la superficie terrestre con el objetivo de registrar el movimiento generado por una fuente energética, como puede ser la detonación de explosivos o el impulso provocado por camiones vibradores¹.

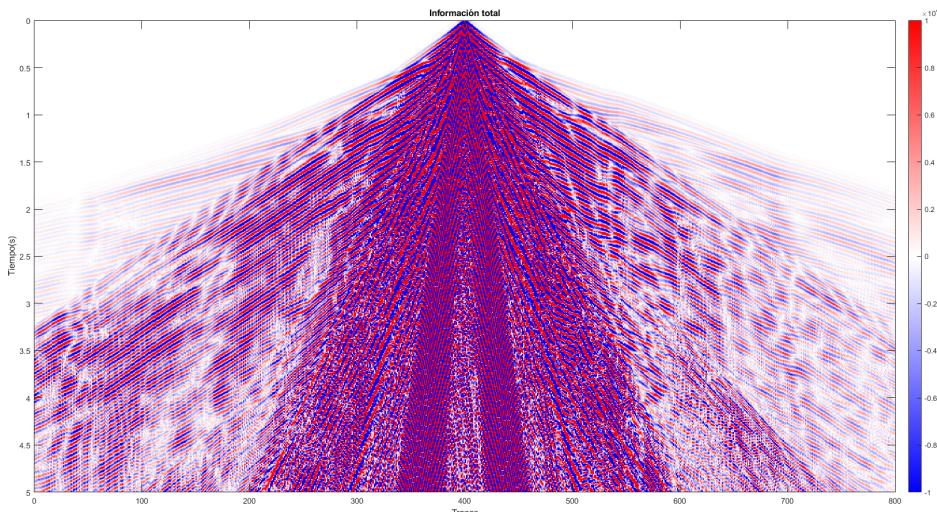


Figura 5.1: Situación idílica en la que obtenemos la señal al completo.

Los geófonos son receptores de señal, encargados de captar las ondas elásticas originadas por las cargas previamente detonadas, y la técnica *Compressed Sensing* se presenta muy aprovechable para

¹De este modo, se obtiene información de las distintas capas que separan la superficie del bruto para posteriormente determinar por dónde se debe realizar la extracción.

conseguir optimizar el proceso de captación y reconstrucción de dicha señal.

En muchas ocasiones, la colocación de geófonos está condicionada por las alteraciones que el terreno presenta, pudiendo existir determinadas zonas en las que sea inviable su colocación y en las que, por lo tanto, no captemos señal. Esto puede deberse a diferencias bruscas en el nivel del terreno o a la existencia de algún obstáculo (como por ejemplo un lago) que imposibilita la colocación de dichos receptores, por lo que es interesante ser capaces de reconstruir la señal a partir de las captaciones que el terreno permite realizar.

Por otro lado, conseguir reducir al máximo el número de detonaciones requeridas para poder registrar una señal fidedigna se traduce en un enorme ahorro económico, siendo este el principal motivo por el que la técnica tiene un enorme potencial en este campo.

Para poder evaluar la técnica, hacemos uso del formato SEG-Y [39], que es uno de los más populares en el campo de geofísica. Cada archivo .sgy contiene una gran cantidad de información referente a la señal captada por los geófonos usando de referencia diferentes dominios². Entre toda esta información, la cual puede corresponder a 5 dominios diferentes, existe una colección de matrices correspondientes al dominio tiempo-geófonos, de las cuales usaremos una como referencia.

De este modo, en la Figura 5.1 se encuentra la información del terreno al completo, simulando una situación en la que la colocación de geófonos se ha podido realizar de manera uniforme, mientras que en la Figura 5.2 forzamos que la señal sea inexistente en distintas zonas **aleatorias** que representan el 50 % de nuestra señal, es decir, la dimensión de captación m es la mitad de la dimensión de la señal original ($m = n/2$), simulando así la ausencia de geófonos en dichas franjas verticales (en adelante trazas).

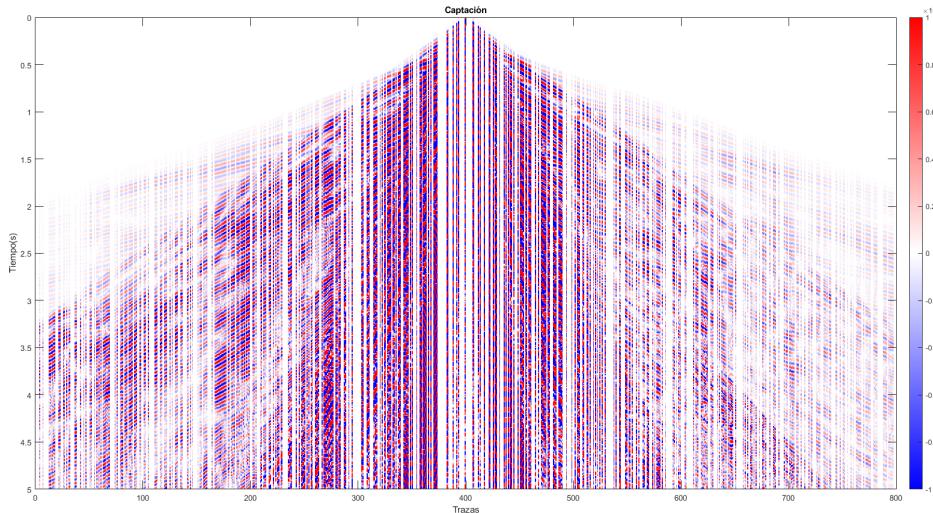


Figura 5.2: Situación en la que solo captamos la mitad de la señal.

Para poder contrastar la reconstrucción, realizamos tres experimentos en paralelo: en el primer caso la matriz de representación está compuesta por coeficientes de Fourier, en el segundo por coeficientes de *Wavelets* de tipo Haar y por último por coeficientes de *Wavelets* de tipo Daubechies. En las

²Esto permite preservar la información de manera fiable aún en condiciones extremas de ruido gracias al fenómeno de redundancia, del cual hablamos en secciones anteriores

tres situaciones se ha optado por una matriz de captación formada por filas aleatorias de la matriz identidad de dimensión acorde al problema, puesto que esto aportará máxima en conjunto con las matrices de representación respectivas, y para proceder a la reconstrucción mediante minimización ℓ_1 seguiremos el algoritmo Basis Pursuit [32].

Para poder mostrar los resultados con mayor claridad, tomaremos pequeños parches de nuestra matriz general para poder observar más de cerca la reconstrucción, aportando finalmente la reconstrucción total en los tres casos.

Los parches se han elegido con el objetivo de cubrir las situaciones más representativas ante las que se puede encontrar la reconstrucción, y en la Figura 5.3 se muestra su posición.

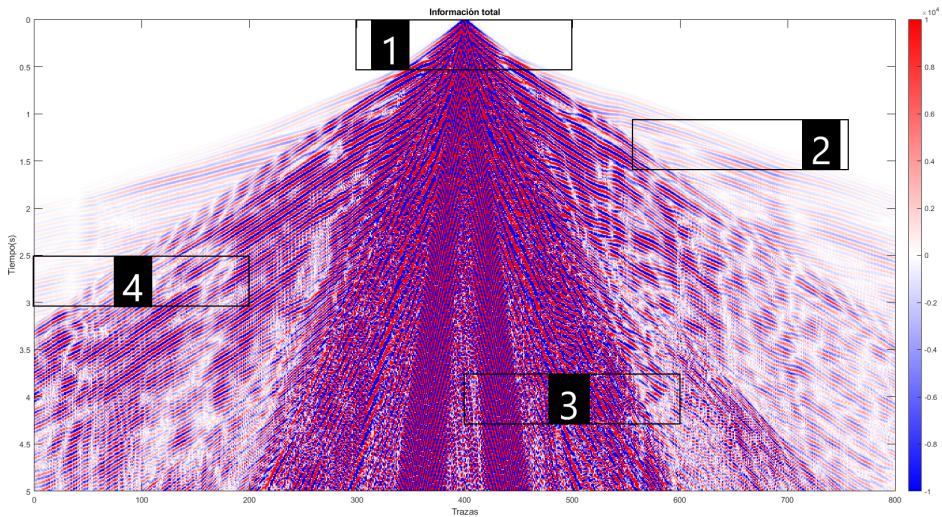


Figura 5.3: Parches sobre los que mostraremos la reconstrucción.

Mientras que la creación de la matriz Θ en el caso unidimensional era un proceso sencillo, en este caso llevará un poco más de elaboración: presentamos a continuación el procedimiento para su composición en el primer caso, en el cual tomaremos como matriz de captación una base de Fourier, concretamente una correspondiente al algoritmo FFT [40]. No obstante, para los casos con *Wavelets* el procedimiento es análogo.

Haremos uso del producto de Kronecker para la creación de una matriz Ψ que permita hacer el cambio de base desde el dominio de frecuencias al dominio temporal teniendo en cuenta que, aunque se trate de un problema en dos dimensiones, realmente podemos proceder como si se tratara de una señal únicamente en función del tiempo puesto que, a diferencia de en la Figura 4.8, la cual fue recuperada partiendo de una muestra aleatoriamente distribuida, en esta ocasión las columnas son captadas en su totalidad, como se muestra en la Figura 5.2. Esto es debido a que en el caso de que un geófono sea colocado en dicha posición, este captará el movimiento ocasionado por la detonación al completo.

Por lo tanto, denotando por X a nuestra señal original, X' a la transformada de Fourier de nuestra señal respecto del eje de ordenadas y por D_j a la matriz de cambio de base del dominio de frecuencias

al dominio temporal³ respecto del eje de ordenadas de la matriz identidad de dimensión $j = 800$ (puesto que la matriz consta de 800 trazas), sabemos que, siguiendo la documentación de SciPy⁴, se verifica la siguiente ecuación:

$$X = X'D_j \quad (5.1)$$

Por lo tanto, siendo I_k la matriz identidad de dimensión $k = 2251$, que es el numero de filas de nuestra matriz X , podemos apoyarnos en la invarianza de la multiplicación por la identidad para considerar una expresión equivalente a (5.1) del tipo:

$$X = I_k X' D_j \quad (5.2)$$

de manera que, haciendo uso de la Ecuación 2.13, tenemos que:

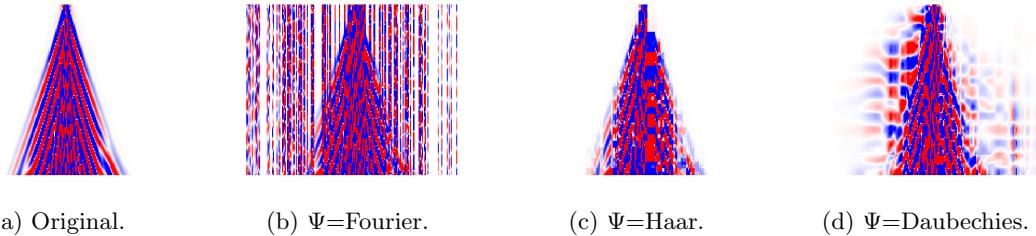
$$(D_j^T \otimes I_k) \text{vec}(X') = \text{vec}(X) \quad (5.3)$$

llegando así a la composición de la matriz $\Psi = D_j^T \otimes I_k$.

Procediendo como en el Ejemplo 4.4.1, tomamos como Φ un número $m = 2251 \cdot 400$ de filas aleatorias de la matriz identidad I_n con $n = 2251 \cdot 800$, llegando de este modo a la composición de Θ para proceder a la reconstrucción.

Presentamos a continuación la reconstrucción de los cuatro parches obtenida en los tres casos. Con carácter general, las matrices correspondientes a los parches tienen unas dimensiones de 200×200 y registran valores que van desde -10000 , valor al cual corresponde el tono azul más oscuro, hasta 10000 (rojo de mayor intensidad):

Figura 5.4: Reconstrucción Parche 1.



Podemos observar en la Figura 5.4 cómo ninguna de las tres reconstrucciones llega a ser del todo satisfactoria, lo cual es esperable si tenemos en cuenta que la zona a reconstruir corresponde a la posición y el momento donde se realiza la detonación, y es por tanto la zona donde se registra mayor oscilación. No obstante, este parche evidencia una diferencia entre las distintas reconstrucciones: mientras que mediante el uso de Wavelets (especialmente los de tipo Haar) podemos determinar con bastante exactitud el momento en el que la detonación alcanza una determinada traza, el uso de coeficientes de Fourier nos priva de esta capacidad, interpolando la traza al completo con coeficientes oscilatorios.

³Esta matriz coincide con el operador análisis introducido en la Proposición 2.3.6

⁴Puede encontrarse en <https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.ifft.html>

Figura 5.5: Reconstrucción Parche 2.

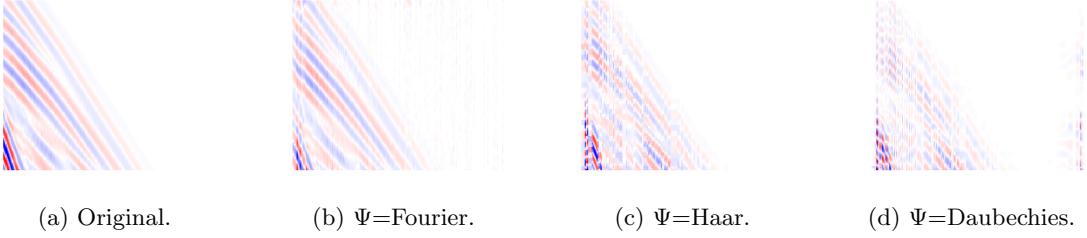


Figura 5.6: Reconstrucción Parche 3.

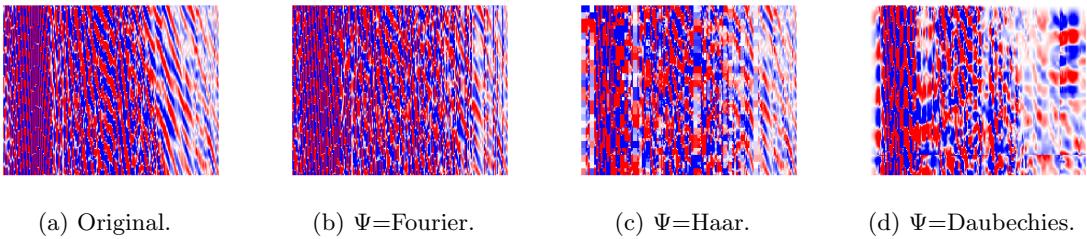
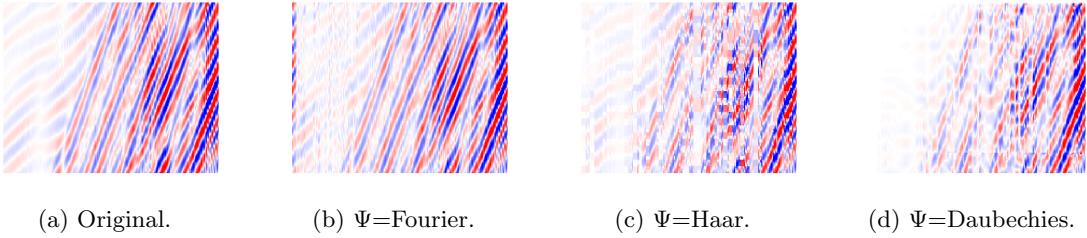


Figura 5.7: Reconstrucción Parche 4.



En los otros tres parches, correspondientes a distintos momentos posteriores a la detonación, podemos percibir como la reconstrucción mediante Wavelets Haar se adapta mejor a zonas en las que las oscilaciones registradas tienen menos recorrido. Este hecho se evidencia, por ejemplo, en la parte izquierda de las subfiguras de la Figura 5.7.

También, se puede apreciar la diferenciabilidad característica de los Wavelets de Daubechies, la cual le lleva a proporcionar una reconstrucción de pocas garantías en zonas donde la oscilación registrada es muy grande, ocasionando la aparición del fenómeno conocido en reconstrucción de señales como *aliasing*⁵. Adicionalmente, dicho procedimiento se muestra incapaz de reconstruir zonas de pequeñas oscilaciones, como se puede observar en la parte izquierda de las subfiguras de la Figura 5.7.

⁵El *aliasing* es el efecto que causa que señales continuas distintas se tornen indistinguibles cuando se muestran digitalmente. Cuando esto sucede, la señal original no puede ser reconstruida de forma única a partir de la señal digital.

Con todo esto, concluimos que la capacidad de reconstrucción de los coeficientes de Fourier para ondas sísmicas es muy superior a la de Wavelets tanto de Daubechies como de Haar, ya que la reconstrucción conseguida mediante FFT es muy cercana al 100 % exitosa hasta en zonas en las que las oscilaciones registradas son máximas (ya que se trata de las trazas más próximas a la zona en la que se detonó la descarga), como es el caso de la Figura 5.6.

No obstante, en los últimos años se ha trabajado en el desarrollo de Wavelets específicos que consigan explotar las características particulares de cada tipo de ondas. En el caso de sísmica, destacan los Curvelets [33] y los Synchrosqueezed Wavelets [34], cuya implementación para el caso estudiado en el presente trabajo sería una interesante línea de trabajo a desarrollar en el futuro.

Presentamos adicionalmente la reconstrucción al completo realizada por los diferentes procedimientos. El proceso resulta ser exitoso computacionalmente hablando en los tres casos, y el coste computacional es factible teniendo en cuenta las dimensiones tan elevadas con las que se está trabajando. En el caso del uso de Wavelets se registra el mayor tiempo de reconstrucción, sobre pasando el proceso al completo los 3 minutos de duración. Esto nos lleva a plantear otra interesante futura línea de trabajo: el desarrollo de algoritmos cuyo consumo de memoria sea inferior al utilizado en nuestros experimentos.

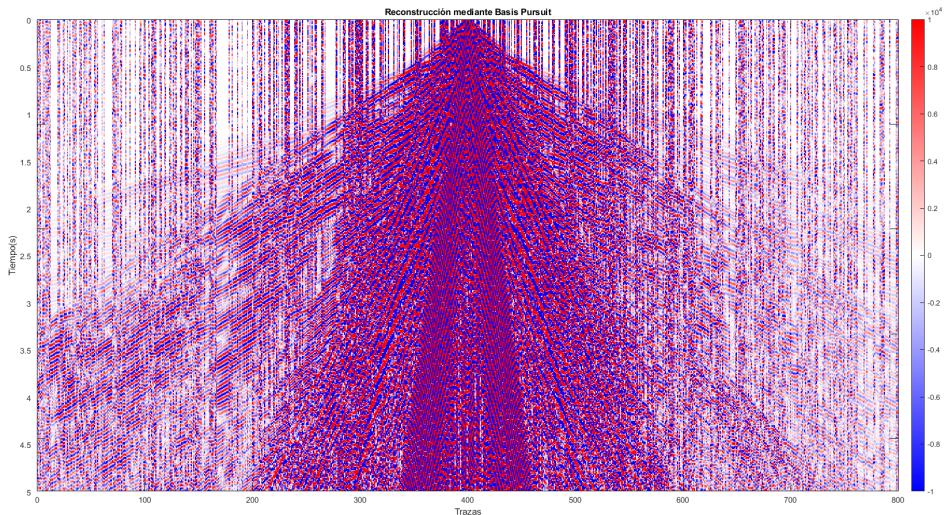


Figura 5.8: Reconstrucción al completo con Ψ formada por coeficientes de Fourier.

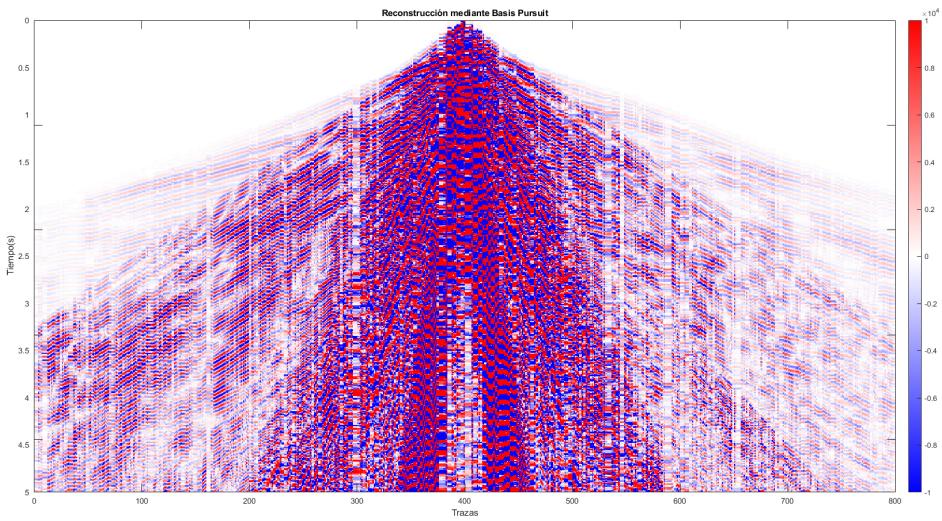


Figura 5.9: Reconstrucción al completo con Ψ formada por coeficientes de Wavelets de tipo Haar.

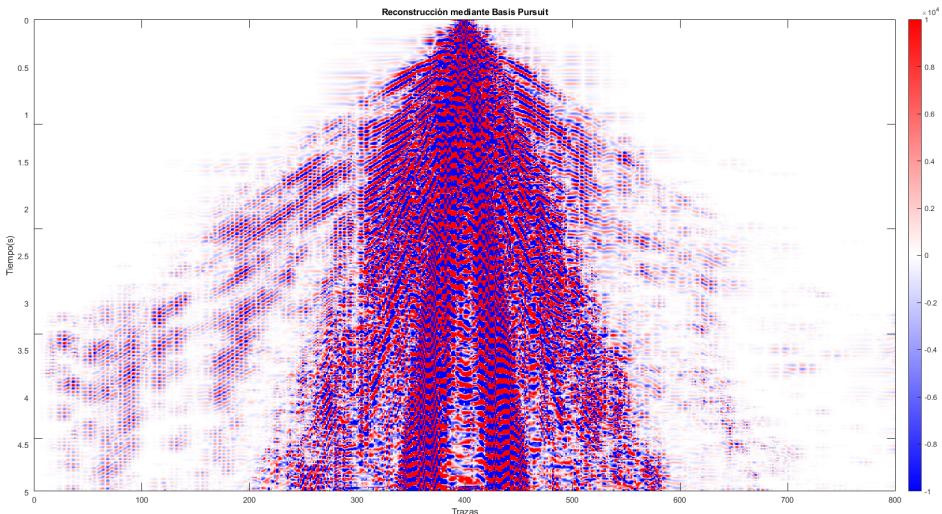


Figura 5.10: Reconstrucción al completo con Ψ formada por coeficientes de Wavelets de tipo Daubechies.

Capítulo 6

Conclusiones

En este trabajo hemos dado una descripción detallada de la técnica de procesamiento de señales *Compressed Sensing*, que permite reconstruir señales con componentes de altas frecuencias siguiendo una tasa de muestreo muy inferior a la que indicaría el Teorema de Nyquist-Shannon.

En los capítulos 1 y 2 se da una introducción a la técnica, presentando los fundamentos matemáticos sobre los que se construirán los conceptos clave en secciones posteriores.

En el capítulo 3 se define el concepto de dispersión, dando ejemplos en una y dos dimensiones de su aprovechamiento con el objetivo de ahorrar memoria en procesos tan cotidianos como mandar archivos de audio o imágenes por vía telemática. Para ello, se introducen los conceptos de transformada de Fourier y su evolución para componentes multifase, los *Wavelets*.

Ya en el capítulo 4 se dan las bases teóricas de la técnica, haciendo énfasis en la naturaleza probabilística de la técnica y del ahorro que permite hacer a la hora de tomar muestras incluso en el caso unidimensional. Para ello, dos procedimientos distintos para reconstruir son presentados, logrando resultados satisfactorios por ambas vías. Una vez hecho esto, se proporciona una reconstrucción de la misma imagen comprimida en el capítulo anterior partiendo de una muestra aleatoria del 10 %.

Por último, el capítulo 5 se dedica a estudiar la aplicación de la técnica a un campo en los que puede proporcionar ahorros importantes, como es el caso de geosísmica. En este campo, la toma de muestras a partir de las cuales proceder a la reconstrucción de una señal tiene un coste muy elevado y el hecho de limitar al máximo su número es de gran interés. Los resultados son satisfactorios, aunque mejorables en cuanto a eficiencia computacional, por lo que posibles vías de mejora también son proporcionadas.

Con el objetivo de que sirva de introducción para los lectores interesados en la técnica, los ejemplos que son presentados a lo largo de todo el trabajo están disponibles en el siguiente repositorio de GitHub, listos para su implementación. También se ha desarrollado un Dockerfile personalizado que permite disponer de todas las librerías empleadas para recrear los ejemplos en una imagen de Docker, disponible en el mismo repositorio bajo el nombre *csDockerfile*:

https://github.com/fragut01/CS_tutorial

Apéndice A

Formulación de un problema de minimización ℓ_1 como un problema de programación lineal

Consideremos el siguiente problema:

- Dados: la matriz A de dimensión $m \times n$ y el vector b de dimensión $m \times 1$
- Encontrar: el vector x de dimensión $n \times 1$ y el vector α de dimensión $m \times 1$ tales que:

$$\begin{array}{ll} \min & \|\alpha\|_1 \\ \text{sujeto a} & Ax + \alpha = b \end{array} \quad (\text{A.1})$$

Este problema puede ser formulado como un problema de programación lineal. Para ello, comenzamos definiendo dos vectores α^+ , α^- de dimensión $m \times 1$ y dos vectores x^+ , x^- de dimensión $n \times 1$ como las partes positiva y negativa de α y x respectivamente¹.

De este modo, podemos reformular el problema (A.1) como sigue:

$$\begin{array}{ll} \min & \alpha^+ + \alpha^- \\ \text{sujeto a} & A(x^+ - x^-) + \alpha^+ - \alpha^- = b \\ \text{siendo} & \alpha^+, \alpha^-, x^+, x^- \geq 0 \end{array} \quad (\text{A.2})$$

Para hacer esta formulación más explícita, definimos los siguientes elementos matriciales formados por cajas:

$$y = \begin{pmatrix} x^+ \\ x^- \\ \alpha^+ \\ \alpha^- \end{pmatrix}$$

¹Del mismo modo que para funciones, definimos las partes positiva y negativa del vector α mediante la aplicación de la siguiente fórmula a las coordenadas α_i :

$$\alpha_i^+ = \max\{\alpha_i, 0\}; \alpha_i^- = \max\{-\alpha_i, 0\}$$

de donde se deduce que $\alpha = \alpha^+ - \alpha^-$ y $|\alpha| = \alpha^+ + \alpha^-$.

$$c = (0 \dots 0 | 0 \dots 0 | 1 \dots 1 | 1 \dots 1)$$

donde las cajas formadas por ceros tienen dimensión $1 \times n$ y las formadas por unos $1 \times m$. Adicionalmente, definiendo I como la matriz identidad de dimensión $m \times m$, podemos formar la matriz B como sigue:

$$B = (A | -A | I | -I)$$

De este modo, el problema (A.2) queda reformulado como:

$$\begin{array}{ll} \min & cy \\ \text{sujeto a} & By = b \\ \text{siendo} & y \geq 0 \end{array} \quad (\text{A.3})$$

que es un problema de minimización de programación lineal.

Bibliografía

Bibliografía principal

- [1] Adam Bowers y Nigel J. Kalton. *An Introductory Course in Functional Analysis*. Springer, 2014.
- [2] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Vol. xx. 2009.
- [3] Peter G. Casazza y Gitta Kutyniok. *Finite Frames. Theory and applications*. Birkhäuser, 2013.
- [4] Mallat Stéphane. «CHAPTER 5 - Frames». En: *A Wavelet Tour of Signal Processing (Third Edition)*. Ed. por Mallat Stéphane. Third Edition. Boston: Academic Press, 2009, págs. 155-204. ISBN: 978-0-12-374370-1. DOI: <https://doi.org/10.1016/B978-0-12-374370-1.00009-4>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123743701000094>.
- [5] Steve L. Brunton y J. Nathan Kutz. *Data Driven Science and Engineering*. 2017. URL: <http://databookuw.com/databook.pdf>.
- [6] Charles Van Loan. *Computational Frameworks for the Fast Fourier Transform*. Society for Industrial y Applied Mathematics, 1992. URL: <https://pubs.siam.org/doi/abs/10.1137/1.9781611970999>.
- [7] D.S. Taubman y Michael Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2002.
- [8] E.J. Candès y M.B. Wakin. «An Introduction To Compressive Sampling». En: *IEEE Signal Processing Magazine* 25.2 (2008).
- [9] Yonina C. Eldar y Gitta Kutyniok. *Compressed Sensing. Theory and applications*. Cambridge, 2012.
- [10] Mark A Davenport. «Random Observations on Random Observations: Sparse Signal Acquisition and Processing». PhD. Rice University, 2010, pág. 204. URL: <https://scholarship.rice.edu/handle/1911/62225>.

Referencias históricas y artículos de consulta

- [11] C. E. Shannon. «A Mathematical Theory of Communication». En: *Bell System Technical Journal* 27.3 (1948). DOI: <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x>.
- [12] H. Nyquist. «Certain Topics in Telegraph Transmission Theory». En: *Transactions of the American Institute of Electrical Engineers* 47.2 (1928).

- [13] R.H. Walden. «Analog-to-digital converter survey and analysis». En: *IEEE Journal on Selected Areas in Communications* 17.4 (1999), págs. 539-550. DOI: 10.1109/49.761034.
- [14] Joseph Fourier. *Theorie analytique de la chaleur*. Chez Firmin Didot, pere et ‘ fils, 1822.
- [15] Huamin Zhang y Feng Ding. «On the Kronecker Products and Their Applications». En: *Journal of Applied Mathematics* 2013 (jun. de 2013). DOI: 10.1155/2013/296185.
- [16] Jorge L. Borges. *Prólogos de la biblioteca de Babel*. Alianza, 2001.
- [17] Alfred Haar. *Zur Theorie der orthogonalen Funktionensysteme*. Mathematische Annalen, 69, 1910, págs. 331-371. DOI: 10.1007/BF01456326.
- [18] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial y Applied Mathematics, 1992. DOI: 10.1137/1.9781611970104. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611970104>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611970104>.
- [19] James W. Cooley y John W. Tukey. *An algorithm for the machine calculation of complex Fourier series*. Mathematics of computation, 1965.
- [20] Ingrid Daubechies. «Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme». En: *SIAM Journal on Mathematical Analysis* 24.2 (1993), págs. 499-519. DOI: 10.1137/0524031. eprint: <https://doi.org/10.1137/0524031>. URL: <https://doi.org/10.1137/0524031>.
- [21] M. R. Garey y D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A Series of Books in the Mathematical Sciences, 1979.
- [22] E. J. Candès y Terence Tao. *The Power of Convex Relaxation: Near-Optimal Matrix Completion*. 2009. arXiv: 0903.1476.
- [23] E. J. Candès, Justin K. Romberg y Terence Tao. «Stable signal recovery from incomplete and inaccurate measurements». En: *Communications on Pure and Applied Mathematics* 59.8 (2006), págs. 1207-1223. DOI: <https://doi.org/10.1002/cpa.20124>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.20124>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20124>.
- [24] Galen Andrew y Jianfeng Gao. «Scalable Training of L1-Regularized Log-Linear Models». En: (2007). URL: <https://www.microsoft.com/en-us/research/publication/scalable-training-of-l1-regularized-log-linear-models/>.
- [25] Robert Malouf. «A Comparison of Algorithms for Maximum Entropy Parameter Estimation». En: *COLING-02: Proceedings of the 6th Conference on Natural Language Learning* 20 (2002).
- [26] Fabian Theis y col. «Signal recovery from partial information via orthogonal matching pursuit». En: *Neural Computation - NECO* 15 (ene. de 2007), págs. 419-439.
- [27] D. Needell y J. A. Tropp. *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*. 2008. arXiv: 0803.2392 [math.NA].
- [28] E. J. Candès y Terence Tao. «Decoding by linear programming». En: *IEEE Transactions on Information Theory* 51.12 (2005).
- [29] P Frankl y H Maehara. «The Johnson-Lindenstrauss lemma and the sphericity of some graphs». En: *Journal of Combinatorial Theory, Series B* 44.3 (1988), págs. 355-362. ISSN: 0095-8956. DOI: [https://doi.org/10.1016/0095-8956\(88\)90043-3](https://doi.org/10.1016/0095-8956(88)90043-3). URL: <https://www.sciencedirect.com/science/article/pii/0095895688900433>.
- [30] R. G. Baraniuk. «Compressive sensing». En: *IEEE Signal Processing Magazine* 24.4 (2007).

- [31] Nasir Ahmed. «How I came up with the discrete cosine transform». En: *Digital Signal Processing* 1.1 (1991), págs. 4-5. ISSN: 1051-2004. DOI: [https://doi.org/10.1016/1051-2004\(91\)90086-Z](https://doi.org/10.1016/1051-2004(91)90086-Z). URL: <https://www.sciencedirect.com/science/article/pii/105120049190086Z>.
- [32] Scott Shaobing Chen, David L. Donoho y Michael A. Saunders. «Atomic Decomposition by Basis Pursuit.» En: *SIAM Rev.* 43.1 (2001), págs. 129-159. URL: <http://dblp.uni-trier.de/db/journals/siamrev/siamrev43.html>.
- [33] Emmanuel J. Candès y David L. Donoho. «New tight frames of curvelets and optimal representations of objects with piecewise C₂ singularities». En: *Communications on Pure and Applied Mathematics* 57.2 (2004), págs. 219-266. DOI: <https://doi.org/10.1002/cpa.10116>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.10116>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.10116>.
- [34] Ingrid Daubechies, Jianfeng Lu y Hau-Tieng Wu. «Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool». En: *Applied and Computational Harmonic Analysis* 30.2 (2011), págs. 243-261. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/j.acha.2010.08.002>. URL: <https://www.sciencedirect.com/science/article/pii/S1063520310001016>.
- [35] E. J. Candès y J. Romberg. «l1-Magic: Recovery of sparse signals via Convex Programming». En: (2005). URL: https://www.researchgate.net/publication/269634028.l1-MAGIC_Recovery_of_Sparse_Signals_via_Convex_Programming.

Paquetes y librerías de Python

- [36] Gregory R. Lee y col. «PyWavelets: A Python package for wavelet analysis». En: *Journal of Open Source Software* 4.36 (2019). DOI: 10.21105/joss.01237. URL: <https://doi.org/10.21105/joss.01237>.
- [37] Steven Diamond y Stephen Boyd. «CVXPY: A Python-embedded modeling language for convex optimization». En: *Journal of Machine Learning Research* 17.83 (2016), págs. 1-5.
- [38] Akshay Agrawal y col. «A rewriting system for convex optimization problems». En: *Journal of Control and Decision* 5.1 (2018), págs. 42-60.
- [39] «SEG-Y r2.0: SEG-Y revision 2.0 Data Exchange format». En: *Society of Exploration Geophysicists* (ene. de 2017).
- [40] Matteo Frigo y Steven G. Johnson. «The Design and Implementation of FFTW3». En: *Proceedings of the IEEE* 93.2 (2005). Special issue on “Program Generation, Optimization, and Platform Adaptation”, págs. 216-231.