



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



Compressed Sensing

Algoritmo Re-Weighted ℓ_1 con pesos actualizados resolviendo un problema dual

Matías Valdés Dini

Programa de Posgrado en Ingeniería Matemática
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Marzo de 2019



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



Compressed Sensing

Algoritmo Re-Weighted ℓ_1 con pesos actualizados resolviendo un problema dual

Matías Valdés Dini

Tesis de Maestría presentada al Programa de Posgrado en Ingeniería Matemática, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magíster en Ingeniería Matemática.

Director:

PhD Prof. Marcelo Fiori Schiavo

Montevideo – Uruguay
Marzo de 2019

Valdés Dini, Matías

Compressed Sensing / Matías Valdés Dini. -
Montevideo: Universidad de la República, Facultad
de Ingeniería, 2019.

XII, 85 p. 29, 7cm.

Director:

Marcelo Fiori Schiavo

Tesis de Maestría – Universidad de la República,
Programa en Ingeniería Matemática, 2019.

Referencias bibliográficas: p. 77 – 82.

1. Sensado Comprimido, 2. Optimización ℓ_1 ,
3. Esparsidad, 4. Weighted ℓ_1 , 5. Multiplicador de
Lagrange, 6. Dualidad, 7. LASSO, 8. Subgradiente.
I. Fiori Schiavo, Marcelo, . II. Universidad de
la República, Programa de Posgrado en Ingeniería
Matemática. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Ph.D. Prof. Diego Armentano

Ph.D. Prof. Pablo Musé

Ph.D. Prof. Ignacio Ramírez

Montevideo – Uruguay
Marzo de 2019

A mi madre y a mi padre
Jóvenes y enamorados
Como el futuro

Agradecimientos

La totalidad de mi formación de maestría en Ingeniería Matemática, de la cual esta tesis pretende ser un reflejo, fue realizada como parte de las horas semanales correspondientes a un cargo docente grado 1, del Instituto de Matemática y Estadística Rafael Laguardia (IMERL), de la Facultad de Ingeniería, Universidad de la República (UdelaR), Uruguay. Agradezco a la dirección del IMERL por brindarme esta posibilidad y a mis compañeros y compañeras de trabajo. Durante la mencionada maestría conté con el apoyo económico de una beca otorgada por la Comisión Académica de Posgrado (CAP) de la UdelaR. Por otro lado, la Comisión Sectorial de Investigación Científica (CSIC) de la UdelaR, financió una beca de movilidad para realizar una estadía académica de un mes en la universidad de Duke, Estados Unidos. El intercambio realizado en dicha estadía, con los integrantes del grupo de investigación del profesor Guillermo Sapiro, fue de mucho provecho para avanzar en el trabajo de tesis. Los experimentos computacionales que se presentan en este trabajo fueron realizados en el Cluster FING, la plataforma de Computación Científica de Alto Desempeño de la UdelaR (www.fing.edu.uy/cluster). Quiero agradecer especialmente a mi director de tesis y director académico, Marcelo Fiori, por la ayuda brindada durante toda la maestría y su buena disposición para discutir ideas en conjunto y permitirme explorarlas.

(Epígrafe:) *"Hay que ponerle pruebas al infinito, para ver si resiste."*

R.J.

RESUMEN

En este trabajo se presentan algunos de los resultados más relevantes de la teoría vinculada al problema de Compressed Sensing (CS) o Sensado Comprimido. Este consiste en: dado un sistema lineal $\Phi x = b$, con infinitas soluciones, hallar una solución con la mayor cantidad de coordenadas nulas posibles. Es decir: la solución más “*esparsa*”. Se propone además una nueva metodología para actualizar los pesos de un algoritmo Re-Weighted ℓ_1 , basada en la relajación lagrangeana, que se traduce en algoritmos con un desempeño comparable al de la metodología usual.

El problema CS resulta de gran interés en la adquisición de señales con características esparsas, como las imágenes y señales de audio. Esto es así pues, mientras que el proceso usual de adquisición realiza n medidas $x^* \in \mathbb{R}^n$ y luego las comprime, CS permite sensar y comprimir x^* en un único paso, a partir de m medidas lineales: $b = \Phi x^*$, con $m \ll n$. Cuando la matriz de medida Φ cumple ciertas propiedades, es posible resolver el problema CS de forma eficiente, recuperando de esta forma la señal esparsa x^* a partir de b . Para esto se resuelve un problema equivalente de optimización convexa, basado en la norma ℓ_1 .

Este proceso de recuperación puede ser mejorado asignando pesos a las coordenadas de la norma ℓ_1 , en un problema convexo conocido como Weighted ℓ_1 . Resolviendo repetidas veces este problema, a la vez que se actualizan los pesos, se obtiene un algoritmo del tipo Re-Weighted ℓ_1 . La metodología de actualización de pesos propuesta en este trabajo consiste en considerar dichos pesos como multiplicadores de Lagrange, pudiendo de esta forma utilizar algoritmos clásicos de la relajación lagrangeana para su actualización.

Palabras claves:

Sensado Comprimido, Optimización ℓ_1 , Esparsidad, Weighted ℓ_1 , Multiplicador de Lagrange, Dualidad, LASSO, Subgradiente.

ABSTRACT

The aim of this work is to present some of the most relevant results from the theory related to the problem of Compressed Sensing (CS). This problem consists of: given a linear system $\Phi x = b$, with an infinite number of solutions, find one with as many null coordinates as possible. That is: select the “sparsest” solution. Furthermore, based on a Lagrange relaxation, a new methodology is introduced for updating the weights of a Re-Weighted ℓ_1 algorithm, which is then translated into algorithms with performance comparable to the usual updating methodology.

The problem of CS is of great interest in the context of acquisition of signals with sparse characteristics, such as images and audio. The reason of this importance is that, while the usual acquisition process does n measures $x^* \in \mathbb{R}^n$ and then compress them, CS makes it possible to sense and compress x^* at the same time, using m linear measurements: $b = \Phi x^*$, with $m \ll n$. When the measurement matrix Φ satisfies some properties, the CS problem can be solved efficiently, recovering in this way the sparse signal x^* from the measures b . This can be done by solving a convex problem based on the ℓ_1 norm.

The recovery process may be enhanced by assigning weights to the ℓ_1 norm coordinates, thus obtaining a convex problem known as Weighted ℓ_1 . By solving this last problem several times, and updating the weights at each step, an algorithm known as Re-Weighted ℓ_1 is obtained. The methodology for updating weights proposed in this work consists in considering those weights as Lagrange multipliers, being able in this way to apply classical Lagrange relaxation algorithms for the update process.

Keywords:

Compressed Sensing, ℓ_1 Optimization, Sparsity, Weighted ℓ_1 , Lagrange Multiplier, Duality, LASSO, Subgradient.

Tabla de contenidos

Lista de símbolos	x
Lista de siglas	x
1 Introducción	1
1.1 Surgimiento y aplicaciones	3
1.2 CS como problema de optimización	4
1.3 Unicidad de soluciones	5
1.4 Complejidad del problema	5
1.5 Minimización ℓ_1	6
1.6 Minimización ℓ_1 con pesos	7
1.7 CS con ruido	9
1.8 Señales compresibles	9
1.9 Generalizaciones de CS	10
1.10 Estructura del documento	11
2 Unicidad de las soluciones de CS	12
2.1 Spark de Φ	12
2.2 Propiedad de Isometría Restringida (RIP)	13
2.3 Coherencia mutua de Φ	15
3 Minimización ℓ_1	17
3.1 Uso de la norma ℓ_1	17
3.2 Problema ℓ_1 sin ruido	19
3.2.1 Equivalencia entre (P_1) y (P_0)	20
3.2.2 Medidas con Φ aleatoria	21
3.2.3 Fenómeno de transición de fase	22
3.2.4 RIP y cambio de base	24
3.3 Problema ℓ_1 con ruido	24
3.3.1 Caso con oráculo	25
3.3.2 Estabilidad respecto al ruido	26
3.3.3 Ruido acotado	26
3.3.4 Ruido Gaussiano	27

3.4	Matrices aleatorias con estructura	28
4	Algoritmos Re-Weighted ℓ_1	29
4.1	Algoritmo $RW\ell_1$ de CWB	29
4.2	Marco teórico del $RW\ell_1$ de CWB	30
4.3	Desempeño bajo ruido acotado	34
4.4	Otros algoritmos RW	34
4.4.1	$RW\ell_2$	34
4.4.2	$RW\ell_2$ regularizado	35
5	Algoritmo $RW\ell_1$ propuesto (sin ruido)	36
5.1	Problema primal con oráculo	36
5.2	Problema relajado	36
5.3	Problema dual	37
5.4	Soluciones del dual	37
5.5	Algoritmo con oráculo (usando método subgradiente proyectado)	39
5.5.1	Método subgradiente proyectado	39
5.5.2	Cálculo del subgradiente	40
5.5.3	Selección del paso	41
5.5.4	Algoritmo propuesto	41
5.6	Algoritmo sin oráculo	42
5.7	Problema primal sin oráculo	43
5.8	Problema relajado sin oráculo	43
5.9	Problema dual sin oráculo	43
5.10	Soluciones del dual	44
6	Algoritmo $RW\ell_1$ propuesto (con ruido)	46
6.1	Problema con oráculo	46
6.2	Problema sin oráculo	47
6.3	Algoritmo propuesto (sin oráculo y con método subgradiente proyectado)	47
7	Implementación: algoritmos para resolver Weighted-ℓ_1	49
7.1	Problema sin ruido: Programación Lineal (LP)	49
7.2	Problema con ruido: Programación Cónica de Segundo Orden (SOCP)	50
7.3	Algoritmos específicos	50
8	Algoritmo RW-LASSO propuesto	52
8.1	Problema Weighted LASSO (W-LASSO)	52
8.2	Algoritmo RW-LASSO con oráculo	53
8.3	Algoritmo RW-LASSO sin oráculo	55

9 Implementación: algoritmos para resolver W-LASSO	57
9.1 Método Gradiente Proximal	57
9.1.1 Gradiente Proximal Acelerado (GPA)	58
9.1.2 Convergencia de GPA	58
9.2 Algoritmo W-FISTA para el problema W-LASSO	59
9.2.1 Convergencia de W-FISTA	60
9.2.2 Condición de parada	60
10 Resultados experimentales (sin ruido)	62
10.1 Modelo utilizado	62
10.2 Valores iniciales y parámetros	63
10.3 Evaluación del desempeño	63
10.4 Cantidad de medidas m fija	63
10.4.1 Elección del parámetro ϵ	63
10.4.2 Desempeño según las iteraciones de RW	65
10.5 Nivel de esparcidad k fijo	66
10.5.1 Desempeño según las iteraciones de RW	66
10.6 $RW\ell_1$ CWB regularizado	68
11 Resultados experimentales (con ruido Gaussiano)	71
11.1 Modelo utilizado	71
11.2 Valores iniciales y parámetros	72
11.3 Evaluación del desempeño	72
11.4 Resultados obtenidos	72
12 Conclusiones finales	75
Referencias bibliográficas	77
Apéndices	83
Apéndice 1 Código utilizado	84

Capítulo 1

Introducción

En este trabajo se analiza el problema de Compressed Sensing (CS), también conocido como Compressed Sampling, el cual consiste en:

Dado un sistema lineal $Ax = b$, con infinitas soluciones, hallar una solución con la mayor cantidad de coordenadas nulas posibles. Es decir, la solución más “*esparsa*” (del inglés “*sparse*”).¹

Para motivar este problema supongamos que se desea almacenar una señal periódica como la de la Figura 1.1 (a), formada por dos sinusoides de frecuencias f y $3f$, para cierto $f > 0$:

$$x(t) = A_1 \sin(2\pi ft) + A_2 \sin(2\pi(3f)t).$$

La Figura 1.1 (b) muestra el espectro de frecuencia de esta señal, que en este caso está formado por dos elementos no nulos, correspondientes a las frecuencias presentes en la misma.

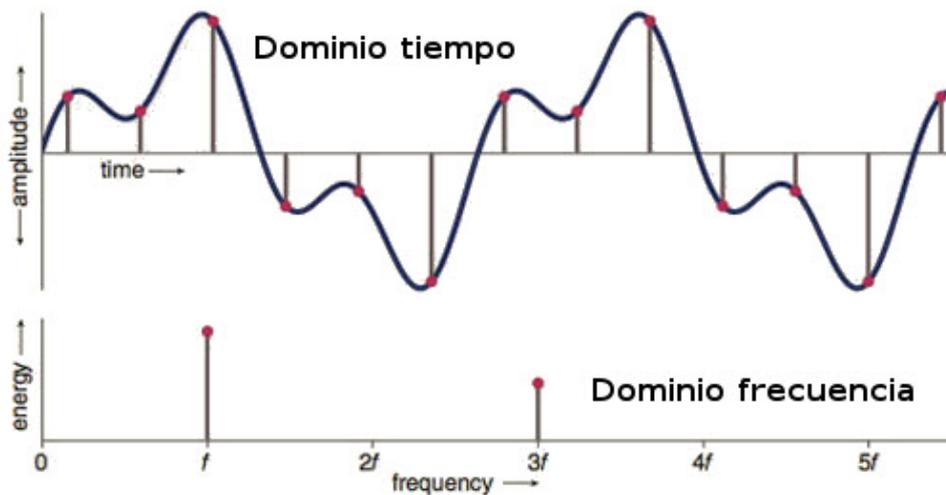


Figura 1.1: (a) Variación temporal de la señal $x(t)$ que se desea adquirir. (b) Componentes de frecuencia de $x(t)$. (Imagen tomada de [40]).

¹En lo que sigue se utiliza como traducción de *sparse* la palabra *esparso*, aunque esta no existe en español. Otras posibles traducciones, utilizando palabras que sí existen en español, pero que no suelen ser empleadas en la literatura de CS, son: escaso, disperso y ralo.

Una forma de adquirir la señal es tomar n muestras equiespaciadas en el tiempo, obteniendo un vector $x^* \in \mathbb{R}^n$. Si la cantidad de muestras por segundo es superior a la tasa de Nyquist, es posible recuperar la señal original a partir de x^* mediante interpolación. Una vez adquiridas estas muestras, buscamos almacenarlas o transmitir las de forma comprimida. Para esto, dado que la señal tiene solamente dos componentes de frecuencia, parece razonable trabajar en el dominio de la frecuencia, y expresar x^* en una base de dicho dominio, obteniendo el vector de frecuencias:

$$\alpha^* = \Psi x^*, \Psi \in \mathbb{R}^{n \times n}.$$

Este vector $\alpha^* \in \mathbb{R}^n$ será *esparso*, con solamente $k = 2$ coordenadas no nulas relevantes, correspondientes a las componentes de frecuencia de $x(t)$. Esto permite comprimir la señal y ahorrar memoria, almacenando solamente $k \ll n$ elementos. Es posible recuperar las muestras originales a partir de α^* mediante: $x^* = \Psi^{-1} \alpha^*$. La Figura 1.2 muestra un esquema de este procedimiento.

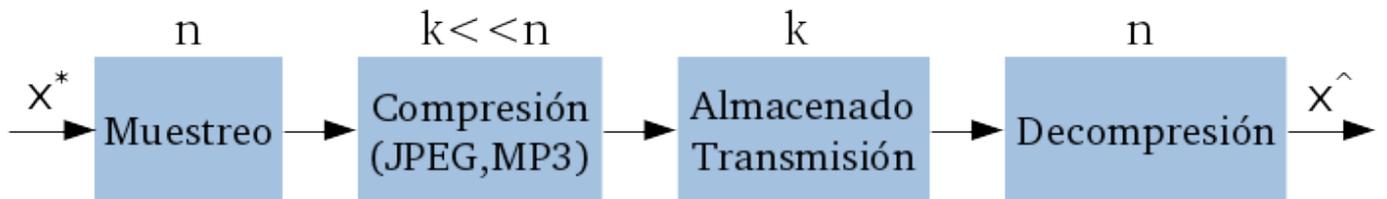


Figura 1.2: Proceso usual para adquirir una señal: muestreo a una tasa superior a la de Nyquist y posterior compresión de las n muestras, almacenando solamente $k \ll n$ elementos.

CS constituye una alternativa al proceso descrito, en la cual el sensado y la compresión se realizan en un único paso; recibiendo por esto el nombre de Compressed Sensing (Sensado Comprimido). Lo que se hace es utilizar una matriz $\Phi \in \mathbb{R}^{m \times n}$, denominada matriz de medida, para obtener m muestras lineales, en la forma: $b = \Phi x^*$. Tomando $m < n$, se reduce la cantidad de muestras realizadas respecto al procedimiento anterior. Sin embargo, esto implica tomar muestras a una tasa inferior a la de Nyquist, por lo que no hay garantías de poder recuperar la señal original. Lo interesante es que, si se añade la hipótesis de que x^* es suficientemente esparso, en muchos casos se lo podrá recuperar a partir de b , aún cuando la cantidad de medidas sea muy pequeña: $m \ll n$. La Figura 1.3 muestra un esquema de este procedimiento. La compresión en el sensado se da cuando se utilizan $m \ll n$ medidas, idealmente $m \simeq k$.

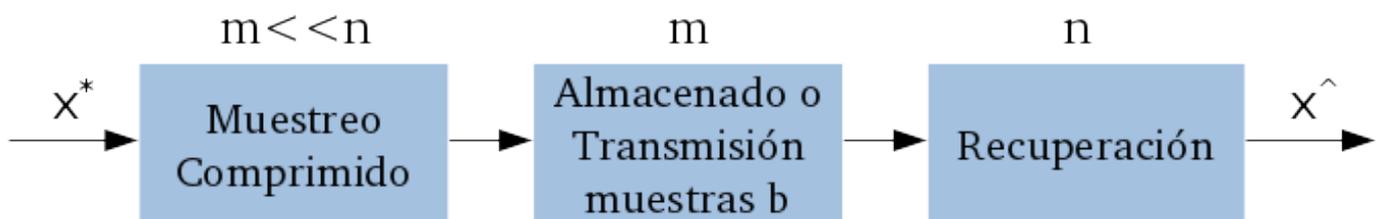


Figura 1.3: Proceso de adquisición de una señal mediante Sensado Comprimido (CS). Si el vector x^* es suficientemente esparso, es posible recuperarlo a partir de muestras lineales, realizadas a una tasa inferior a la de Nyquist: $m \ll n$.

En el ejemplo considerado el vector esparso se obtiene al transformar x^* al dominio de la frecuen-

cia. Se busca entonces la solución más esparsa del sistema $\Phi\Psi^{-1}\alpha = b$, con medidas $b = \Phi x^*$. De esta forma se obtiene un problema CS, con matriz $A = \Phi\Psi^{-1}$. Salvo que se indique lo contrario, en lo que sigue se asume que las señales de interés son esparsas en la base canónica de \mathbb{R}^n . Es decir: $\Psi = Id$.

Otro contexto en el cual se obtiene el problema CS es en “*Sparse Modeling*” (modelado esparsa). Haciendo una descripción muy general, en este se tiene una señal b de alta dimensión (por ejemplo una imagen), y se la desea modelar como una superposición esparsa de cierto tipo de “señales”, dadas por las columnas de una matriz A , denominada diccionario:

$$b = Ax = \sum_{i=1}^n A^i x_i, \quad x \text{ esparsa.}$$

Si por ejemplo se desea modelar una señal $b \in \mathbb{R}^m$ como una superposición esparsa de sinusoides y deltas, se puede considerar el diccionario $A = [F, I] \in \mathbb{R}^{m \times 2m}$, donde $F \in \mathbb{R}^{m \times m}$ es la matriz de Fourier, cuyas columnas forman una base de sinusoides, e $I \in \mathbb{R}^{m \times m}$ la matriz identidad, que conforma una base de deltas. Se busca entonces un vector de coeficientes esparsa $x \in \mathbb{R}^{2m}$ y tal que:

$$b = Ax = \sum_{i=1}^m F^i x_i + \sum_{i=m+1}^{2m} I^i x_i.$$

Una introducción al Sparse Modeling se puede encontrar en [13].

1.1. Surgimiento y aplicaciones

La teoría de CS surge como tal de forma muy reciente, con los trabajos de Donoho [33] y de Candès, Romberg y Tao [18, 17]. Candès se interesó por el problema en febrero de 2004, mientras intentaba recuperar una imagen médica afectada por ruido, conocida como “Shepp–Logan phantom” [36]. Desde entonces ha habido un gran desarrollo de esta teoría, motivado en particular por sus aplicaciones. Estas abarcan diversas áreas como: imagenología medica [49, 45], conversión analógico/digital [68], detección mediante RADAR [4, 42], corrección de errores [18], cámaras digitales [34, 21, 51], Sismología [41], Radioastronomía [71], comunicación inalámbrica [63, 3, 65] y redes de sensores inalámbricos [48, 47]. En [59] se puede encontrar un análisis más exhaustivo de las aplicaciones.



(a) full sampling



(b) 39% sampling,
SNR=32.2

Figura 1.4: Uso de CS en una Resonancia Magnética (MRI) de pélvis. (a) Imagen de MRI usual (b) MRI obtenida mediante CS con un 61 % menos de muestras, reduciendo así el tiempo de exposición del paciente. (Imagen tomada de [72]).

Si bien en la literatura se considera que la teoría de CS surge como tal con los trabajos publicados entre 2004 y 2006, muchas de las metodologías y herramientas matemáticas utilizadas eran conocidas desde hacía algunas décadas. La minimización ℓ_1 , utilizada en la etapa de reconstrucción de la señal, ya había sido aplicada para recuperar señales esparsas en la década de 1970, en el área de Sismología Reflectiva [24, 64]. Entre mediados de 1980 y principios de 1990 se mejora esta técnica de reconstrucción [61], y Donoho, Stark y Logan obtienen los primeros resultados relevantes sobre su desempeño teórico [31, 30]. Desde la década de 1990, la minimización ℓ_1 también fue utilizada para la reconstrucción de imágenes con gradiente esparso, en un problema de optimización conocido como “Total Variation” (TV) [60, 9]. En forma paralela, en 1993 Mallat y Zhang introducen un algoritmo de reconstrucción de señales esparsas, alternativo a la minimización ℓ_1 , conocido como Matching Pursuit (MP) [50]. Ese mismo año Pati, Rezaifar y Krishnaprasad, introducen mejoras en MP, obteniendo el algoritmo Orthogonal Matching Pursuit (OMP) [58]. Variantes de este algoritmo son ampliamente utilizadas en la actualidad para reconstruir la señal de forma eficiente.

1.2. CS como problema de optimización

Dado un vector de medidas $b = \Phi x^*$, el problema CS asociado resulta equivalente al siguiente problema de optimización:

$$\begin{aligned} \text{argmín} \quad & \|x\|_0. & (P_0) \\ \Phi x &= b \\ x &\in \mathbb{R}^n \end{aligned}$$

Donde la pseudo-norma ℓ_0 cuenta la cantidad de coordenadas no nulas del vector: ¹

$$\|x\|_0 := \#\{i / x_i \neq 0\}.$$

1.3. Unicidad de soluciones

Si bien el Problema (P_0) , equivalente a CS, siempre tiene alguna solución, se busca que esta sea única, de forma que al resolverlo se pueda recuperar la señal x^* de interés de forma unívoca. Para formalizar este requisito se introducirán algunas definiciones. Se dice que una señal es k -esparsa cuando tiene a lo sumo k elementos no nulos: $\|x\|_0 \leq k$. El conjunto formado por estas señales es:

$$\Sigma_k := \{x \in \mathbb{R}^n / \|x\|_0 \leq k\}.$$

Se busca entonces que, para toda señal $x^* \in \Sigma_k$, el sistema $\Phi x = b$, con medidas $b = \Phi x^*$, tenga solución k -esparsa única. Como se verá más adelante, para garantizar este tipo de unicidad será necesario disponer de al menos $2k$ medidas: $m \geq 2k$. Por otro lado, resulta conveniente poder recuperar cualquier x^* utilizando la menor cantidad de medidas m . Existe entonces un compromiso entre ambos requisitos: tomar m pequeño para ahorrar medidas y a la vez utilizar m suficientemente grande para lograr unicidad.

1.4. Complejidad del problema

Además de poder garantizar que (P_0) tiene solución única, es necesario disponer de un algoritmo para hallarla. Un posible algoritmo es el siguiente:

- 1) Buscar soluciones 1-esparsas: $x^* \in \Sigma_1$, probando con todas las posibles formas de elegir el soporte de x^* .
- 2) Si no existen tales soluciones, buscar $x^* \in \Sigma_2$, probando con todos los posibles soportes de dos elementos.
- ⋮
- l) En el paso l de este algoritmo, en que se busca una solución $x^* \in \Sigma_l$, se tienen C_l^n posibles soportes S . Para cada uno, se busca alguna solución del sistema sobredeterminado: $\Phi_S x = b$, donde Φ_S se compone de las columnas de Φ con índices en S .

Si bien este algoritmo es válido, y siempre obtiene una solución de (P_0) , en la práctica no resulta útil. Esto se debe a que, si $x^* \in \Sigma_k$, en el peor caso se tienen $\sum_{i=1}^k C_i^n$ posibles subconjuntos con los cuales probar, resultando en una complejidad combinatoria, por lo que la ejecución podría llevar demasiado tiempo.

¹Esta función no es una norma pues no cumple la propiedad homogénea: $\|\alpha x\|_0 \neq |\alpha| \cdot \|x\|_0$.

Por ejemplo: para $n = 100$ y $k = 15$, aún en el caso ideal en que se conoce el valor de k , se tienen $C_{15}^{100} \simeq 2.53 \times 10^{17}$ posibles soportes. Si cada prueba demanda $1\mu s$, el algoritmo podría demorar unos 8000 años en hallar una solución.

Si bien es posible encontrar algoritmos más eficientes, no existe un algoritmo que permita resolver (P_0) en general, con tiempo de ejecución polinomial en las dimensiones del problema. Esto es debido a que CS se encuentra dentro de los problemas de tipo NP-difícil [38][Sección 2.3]. Surge entonces una limitante importante en el uso de CS para las aplicaciones mencionadas anteriormente, las cuales requieren tiempos de ejecución razonablemente bajos.

1.5. Minimización ℓ_1

Debido a la complejidad de (P_0) en el caso general, se han desarrollado algoritmos que buscan resolver casos particulares del mismo, donde Φ cumple ciertas propiedades. Una primer opción, que no será analizada en este trabajo, es utilizar algoritmos de tipo “Greedy”, entre los que destacan: “Orthogonal Matching Pursuit (OMP)” [58], OMP Regularizado (ROMP) [54], “Iterative hard thresholding (IHT)” [10] y “Compressive Sampling Matching Pursuit” (CoSaMP) [53]. Se puede encontrar una presentación mas detallada de estos en [35][Capitulo 8]. Una segunda opción, que será la considerada en este trabajo, consiste en reemplazar (P_0) por otro problema de optimización pero de más fácil resolución. Usualmente la pseudo-norma ℓ_0 es reemplazada por la norma ℓ_1 , resultando:

$$\begin{aligned} \operatorname{argmín} \quad & \sum_{i=1}^n |x_i|. \\ \Phi x = b \\ x \in \mathbb{R}^n \end{aligned} \quad (P_1)$$

El problema (P_1) se conoce como “Basis Pursuit” [23] y tiene la ventaja de ser convexo, por lo que se puede obtener una solución del mismo de forma eficiente. Además de esto, para cierto tipo de matrices de medida Φ , es posible garantizar que la solución de (P_1) coincide con la solución x^* de (P_0) , cualquiera sea la señal $x^* \in \Sigma_k$ considerada. Es decir que en esos casos es posible recuperar cualquier señal k -esparsa resolviendo (P_1) . En particular se verá que esto es válido, con alta probabilidad, cuando las medidas se realizan con Φ aleatoria de distribución Gaussiana.

Existen, sin embargo, casos en que (P_0) y (P_1) no son equivalentes. Esto es: (P_1) tiene soluciones que no son solución de (P_0) , o peor aún: la solución buscada de (P_0) no se encuentra dentro del conjunto de soluciones de (P_1) . Un ejemplo de esto último se puede obtener tomando [20][Sección 2.1]:

$$\Phi = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

El sistema $\Phi x = b$ tiene infinitas soluciones, siendo la más esparsa: $x^* = (0, 1, 0)$. En la Figura 1.5 (a) se muestra x^* y la recta formada por las soluciones de $\Phi x = b$. Se muestra además la bola ℓ_1 centrada en el origen y de radio $\|x^*\|_1$. Tal como se observa en la Figura 1.5 (b), el sistema lineal

admite además la solución menos esparsa: $\hat{x} = (\frac{1}{3}, 0, \frac{1}{3})$. Como:

$$\|\hat{x}\|_1 = \frac{2}{3} < 1 = \|x^*\|_1,$$

se concluye que la solución buscada x^* de (P_0) no es solución del problema convexo (P_1) .

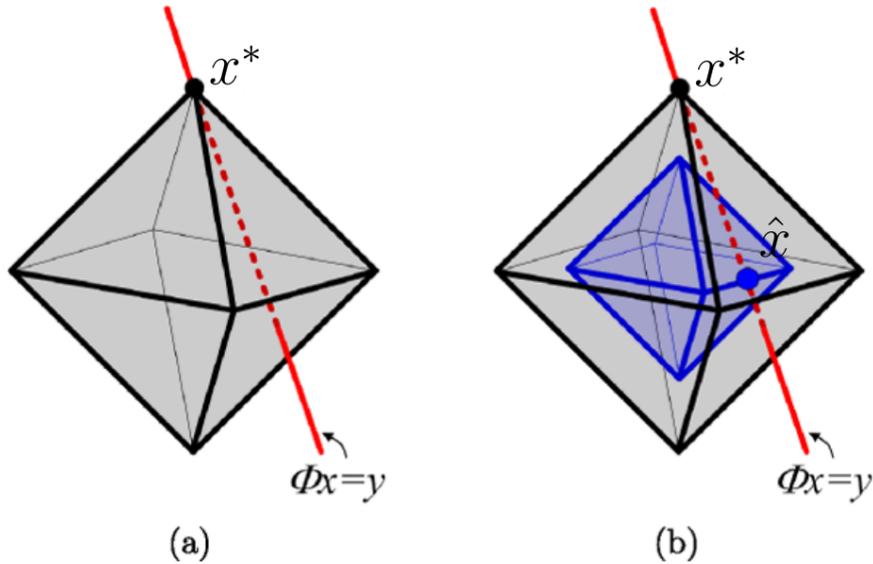


Figura 1.5: (a) x^* solución buscada de (P_0) , bola ℓ_1 de radio $\|x^*\|_1$ (en gris) y recta formada por las soluciones de $\Phi x = b$ (en rojo); (b) Existe \hat{x} , solución de $\Phi x = b$, con $\|\hat{x}\|_1 = \frac{2}{3} < 1 = \|x^*\|_1$. Por lo tanto x^* no es solución del problema convexo (P_1) . (Imagen tomada de [20]).

1.6. Minimización ℓ_1 con pesos

Es posible evitar lo anterior agregando pesos no negativos a las coordenadas de la norma ℓ_1 ; obteniendo así el problema conocido como “Weighted ℓ_1 ” (ℓ_1 con pesos):

$$\begin{aligned} \operatorname{argmín} \quad & \sum_{i=1}^n w_i |x_i|. & (P_1W) \\ \Phi x = b & \\ x \in \mathbb{R}^n & \end{aligned}$$

Si se eligen los pesos w_i de forma adecuada, las soluciones de (P_1W) serán solución de (P_0) . En el ejemplo anterior se puede lograr esto tomando por ejemplo: $w = (3, 1, 3)$. Geométricamente, los pesos eliminan la simetría de la bola ℓ_1 , como se puede apreciar en la Figura 1.6 (b). En general, en el caso ideal en que se conoce una solución x^* de (P_0) , se puede tomar:

$$w_i = \begin{cases} \frac{1}{|x_i^*|} & \text{si } x_i^* \neq 0 \\ \infty & \text{si } x_i^* = 0 \end{cases}.$$

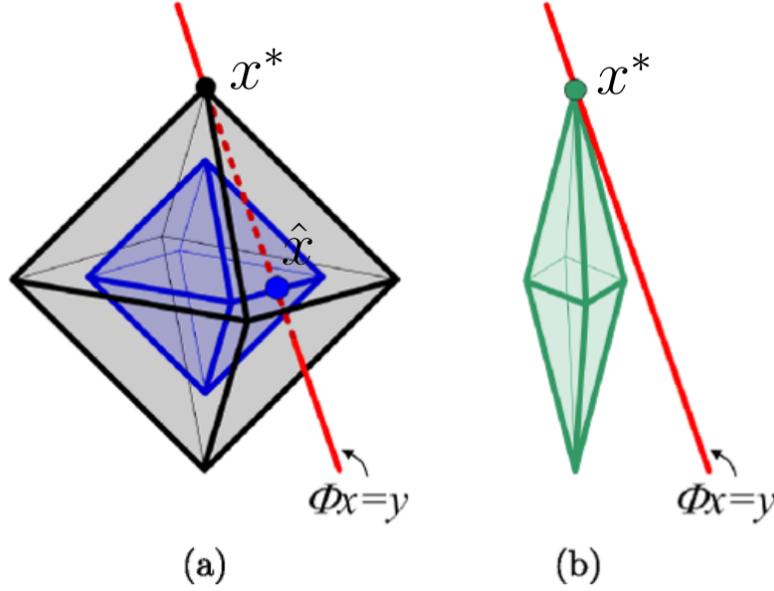


Figura 1.6: (a) Existe \hat{x} , solución de $\Phi x = b$, con $\|\hat{x}\|_1 < \|x^*\|_1$. Por lo tanto la señal de interés x^* no es solución del problema convexo (P_1) ; (b) Bola ℓ_1 con pesos $w = (3, 1, 3)$. No existe $\hat{x} \neq x^*$, solución de $\Phi x = b$, y tal que: $\|W\hat{x}\|_1 \leq \|Wx^*\|_1$. Por lo tanto x^* es la única solución del problema convexo (P_1W) . (Imagen tomada de [20]).

En el caso general, asumiendo $w_i \geq 0$, el problema (P_1W) resulta convexo, por lo que se puede obtener una solución del mismo de forma eficiente. La dificultad radica en seleccionar los pesos de forma adecuada, logrando que las soluciones de (P_1W) coincidan con las de (P_0) . En [20], Candès, Wakin y Boyd proponen un algoritmo iterativo para estimar los pesos, partiendo de $w^0 = \vec{1}$ y actualizando sus coordenadas mediante:

$$w_i^{k+1} = \frac{1}{|x_i^{k+1}| + \epsilon}, \quad \epsilon > 0.$$

Donde:

$$x^{k+1} \in \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\operatorname{argmín}} \sum_{i=1}^n w_i^k |x_i|.$$

Dado que en cada paso es necesario resolver un problema Weighted ℓ_1 , el algoritmo se conoce como Re-Weighted ℓ_1 ($RW\ell_1$). En este trabajo se propone una nueva metodología para actualizar los pesos de un algoritmo $RW\ell_1$, basada en considerar dichos pesos como los multiplicadores de Lagrange de un problema convexo. Esto permite poner el problema (P_1W) dentro del marco teórico bien conocido de la relajación Lagrangeana. En particular permite estimar los pesos utilizando algoritmos destinados a estimar multiplicadores de Lagrange.

1.7. CS con ruido

En la práctica las medidas suelen estar afectadas por ruido, por lo que la formulación del problema CS debe ajustarse para tener esto en cuenta. En el caso en que el ruido z es aditivo: $b = \Phi x + z$, se busca resolver, para cierto nivel de ruido $\eta \geq 0$, el problema:

$$\begin{aligned} \operatorname{argmín} \quad & \|x\|_0. & (P_0^\eta) \\ \|\Phi x - b\|_2 & \leq \eta \\ x & \in \mathbb{R}^n \end{aligned}$$

Las fuentes de ruido son diversas. Por lo general los sensores, al estar formados por componentes eléctricos, se ven afectados por ruido térmico, intrínseco a los mismos. Por otro lado, en el proceso de almacenamiento de datos en una computadora, se introduce un error debido a la resolución limitada del sistema de representación de la máquina, que también puede ser modelado como ruido. Como se verá más adelante, para resolver (P_0^η) con un algoritmo que sea “estable” respecto al ruido, ya no será suficiente contar con $m \geq 2k$ medidas como en el caso sin ruido, sino que se requerirá:

$$m \geq Ck \ln \left(e \frac{n}{k} \right).$$

El problema (P_0^η) también resulta NP-difícil en general, para cualquier $\eta \geq 0$ [38][Sección 2.3]. Por este motivo, al igual que en el caso sin ruido, se considera el problema alternativo en el cual la pseudo-norma ℓ_0 se reemplaza por la norma ℓ_1 :

$$\begin{aligned} \operatorname{argmín} \quad & \|x\|_1. & (P_1^\eta) \\ \|\Phi x - b\|_2 & \leq \eta \\ x & \in \mathbb{R}^n \end{aligned}$$

En este caso los algoritmos de tipo RW ℓ_1 mantienen la misma forma, salvo que en cada paso se debe resolver un problema Weighted ℓ_1 con ruido:

$$\begin{aligned} \operatorname{argmín} \quad & \sum_{i=1}^n w_i |x_i|. & (P_1^\eta W) \\ \|\Phi x - b\|_2 & \leq \eta \\ x & \in \mathbb{R}^n \end{aligned}$$

1.8. Señales compresibles

Además de considerar el ruido, se debe tener en cuenta que las señales de interés no suelen ser exactamente k -esparsas. De todas formas, al ser expresadas en una base adecuada, muchas de estas pueden ser bien aproximadas por una señal k -esparsa. Por este motivo, los resultados de CS se suelen expresar en función de la distancia al conjunto de señales k -esparsas Σ_k , dada por:

$$\sigma_k(x)_q := \min_{\hat{x} \in \Sigma_k} \|x - \hat{x}\|_q, \quad \forall q \geq 1.$$

Claramente una señal es k -esparsa si y sólo si $\sigma_k(x)_q = 0$. Dentro de las señales que no son k -esparsas resultan de particular interés las denominadas “compresibles”, donde la distancia a Σ_k decrece exponencialmente con k . Formalmente un vector $x \in \mathbb{R}^n$ se dice compresible, con constante C y tasa $r > 0$, si [35][Sección 1.3]:

$$\sigma_k(x)_2 \leq \frac{C}{k^r}, \forall k = 1, \dots, n.$$

En forma equivalente, una señal se dice compresible, si la magnitud de sus coordenadas, reordenadas en forma decreciente en el vector \hat{x} , decrecen exponencialmente: [35][Sección 1.3]:

$$|\hat{x}_i| \leq \frac{\hat{C}}{i^s}, \forall i = 1, \dots, n.$$

En las definiciones anteriores, referidas a señales compresibles, se ha asumido que la base utilizada para representar a x es la canónica. Sin embargo, es importante tener presente que una señal puede ser compresible o no según la base Ψ en la cual se la represente. Por ejemplo, para señales de audio puede ser útil una base de tipo Fourier, como la Transformada Discreta Coseno (DCT) [1], y para las imágenes una base de funciones de tipo “Wavelets” [2].

1.9. Generalizaciones de CS

El problema CS admite al menos dos generalizaciones, las cuales no serán consideradas en el resto de este trabajo, pero resulta interesante mencionar. Una primer generalización está vinculada al denominado “Problema de Minimización del Rango” (RMP). Este último consiste en hallar la matriz $X \in \mathbb{R}^{m \times n}$ de menor rango entre todas las que cumplen cierta propiedad C [37]:

$$\underset{X \in C \subset \mathbb{R}^{m \times n}}{\operatorname{argmín}} \quad rg(X) = \underset{X \in C \subset \mathbb{R}^{m \times n}}{\operatorname{argmín}} \quad \|\sigma_X^2\|_0, \quad (\text{RMP})$$

donde σ_X^2 es el vector conformado por los valores singulares de la matriz X . El RMP presenta diversas aplicaciones, entre las que destacan el problema de “completar” una matriz (“matrix completion”) [15]. Es fácil ver que el problema CS es un caso particular del problema RMP. Para esto basta considerar el subconjunto de las matrices diagonal: $X = \operatorname{diag}(x)$, y notar que estas verifican:

$$rg(X) = \|x\|_0.$$

Un segundo tipo de generalización de CS consiste en reemplazar la condición de que la señal buscada es k -esparsa, por la hipótesis de que dicha señal pertenece a una variedad de dimension k [5, 70]. Esta perspectiva se basa en que, aunque la señal se genere en un espacio de dimension n , la información relevante de la misma se encuentra en un espacio de dimension $k \ll n$, por lo que resulta conveniente trabajar directamente en dicho espacio.

1.10. Estructura del documento

El resto del documento se encuentra organizado de la siguiente forma. En el Capítulo 2 se dan algunos resultados que permiten garantizar unicidad de soluciones del problema (P_0) en el caso en que las medidas no están afectadas por ruido. En el Capítulo 3 se brindan condiciones que garantizan la equivalencia entre el problema convexo (P_1) , denominado Basis Pursuit, y (P_0) . Se considera el caso de medidas con y sin ruido. Se introducen además las matrices de medida con entradas aleatorias de distribución Gaussiana, que serán utilizadas en el resto del trabajo. En el Capítulo 4 se analiza el problema ℓ_1 con pesos, conocido como Weighted ℓ_1 , y el algoritmo de tipo $RW\ell_1$ propuesto por Candès, Wakin y Boyd (CWB) en [20]. Se analiza además el desempeño de este algoritmo en el caso en que las medidas están afectadas por ruido acotado. En el Capítulo 5 se introduce la nueva metodología $RW\ell_1$ propuesta en el caso ideal, o con oráculo, en que se conoce una solución de (P_0) , y para medidas sin ruido. A partir de esta metodología se obtiene un algoritmo de tipo $RW\ell_1$ basado en el “método subgradiente proyectado”. Se considera luego su extensión al caso no ideal, o sin oráculo, en que no se conoce una solución de CS. En el Capítulo 6 se extiende la metodología presentada en el Capítulo 5 para el caso en que las medidas presentan ruido aditivo. En el Capítulo 7 se discute cómo estimar soluciones de un problema Weighted ℓ_1 genérico. Esto resulta fundamental para la implementación de los algoritmos $RW\ell_1$ ya que estos deben hallar soluciones de un problema Weighted ℓ_1 en cada iteración. En el Capítulo 8 se presenta el problema W-LASSO y se aplica la misma metodología presentada para Weighted ℓ_1 , junto con el método subgradiente, para obtener un algoritmo de tipo RW-LASSO. En el Capítulo 9 se presenta un algoritmo conocido como Weighted Fast ISTA, que permite resolver el problema W-LASSO de forma eficiente. En los Capítulos 10 y 11 se realizan simulaciones para analizar el desempeño de los algoritmos RW propuestos, en el caso sin y con ruido respectivamente, y en relación al algoritmo $RW\ell_1$ de CWB. Finalmente en el Capítulo 12 se presentan las conclusiones finales del trabajo.

Capítulo 2

Unicidad de las soluciones de CS

En este capítulo se resumen algunos de los principales resultados conocidos sobre la unicidad de soluciones del problema CS sin ruido (P_0). Como fuera mencionado en la introducción, es necesario que (P_0) tenga solución única, pues de lo contrario no podríamos recuperar la señal de interés de forma unívoca. Resulta entonces conveniente dar la siguiente definición.

Definición 1. *Dada $\Phi \in \mathbb{R}^{m \times n}$, diremos que hay k -unicidad de soluciones de CS si: **para todo** vector de medidas $b = \Phi x^*$, con $x^* \in \Sigma_k$, el sistema asociado $\Phi x = b$ tiene solución k -esparsa única.*

Notar que no basta con pedir que para algún vector de medidas $b = \Phi x^*$, con $x^* \in \Sigma_k$, el problema (P_0) asociado tenga solución única; sino que se debe tener unicidad **para todo** b generado de esa forma. Esto es debido a que la x^* de interés podría ser cualquier señal k -esparsa.

2.1. Spark de Φ

Una primer caracterización de la k -unicidad de (P_0) tiene que ver con el núcleo de la matriz Φ .

Teorema 1. (*[35][Sección 1.4]*) *Se tiene k -unicidad si y solo si $N(\Phi) \cap \Sigma_{2k} = \{\vec{0}\}$.*

Es decir que no pueden haber vectores $2k$ -esparsos en el núcleo de Φ . El resultado anterior puede ser expresado en términos de una propiedad de la matriz, denominada “spark”, introducida por Donoho y Elad en [29].

Definición 2. (*[35][Definición 1.1]*) *Se define el “spark” de una matriz Φ como la menor cantidad de columnas de Φ linealmente dependientes.*

El spark de Φ está vinculado a su núcleo de la siguiente forma [46][Sección 3]:

$$\text{spark}(\Phi) = \min\{k / N(\Phi) \cap \Sigma_k \neq \{\vec{0}\}\}.$$

Por lo tanto el resultado anterior se puede expresar en términos del spark como:

Teorema 2. (*[35][Teorema 1.1]*) *Se tiene k -unicidad si y sólo si $\text{spark}(\Phi) > 2k$.*

Las matrices de CS cumplen $m \leq n$, por lo que no podrán tener mas de m columnas linealmente independientes. Es decir que cualquier conjunto formado por $m+1$ columnas de Φ debe ser linealmente dependiente y por lo tanto: $\text{spark}(\Phi) \leq m + 1$. Esto lleva al siguiente corolario.

Corolario 1. *Es necesario realizar al menos $2k$ medidas para garantizar k -unicidad: $m \geq 2k$.*

El resultado anterior establece una cota inferior en la cantidad de medidas necesarias para garantizar k -unicidad, la cual resulta proporcional a la cantidad de “información” k de la señal. En la práctica la cantidad de medidas suficientes dependerá de la matriz de medida Φ utilizada. Si se permite elegir esta matriz, la cota puede ser alcanzada y basta con disponer de $m = 2k$ medidas, como lo muestra el siguiente teorema.

Teorema 3. *([38][Teorema 2.14]) Sea $n \geq 2k$ dado. Existe una matriz $\Phi \in \mathbb{R}^{m \times n}$, con $m = 2k$ filas, que garantiza k -unicidad.*

Una forma de construir esta matriz es mediante una matriz de Vandermonde (traspuesta), con $0 < t_1 < \dots < t_n$:

$$\Phi = \begin{pmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_n \\ \vdots & \vdots & \ddots & \vdots \\ t_1^{2k-1} & t_2^{2k-1} & \dots & t_n^{2k-1} \end{pmatrix} \in \mathbb{R}^{2k \times n}.$$

Se dispone entonces de una matriz de medida Φ que permite recuperar cualquier señal k -esparsa con el mínimo de medidas lineales: $m = 2k$. Para esto basta con resolver el problema asociado (P_0), que tendrá solución única. Sin embargo, al intentar utilizar esta estrategia surgen al menos dos limitantes prácticas:

1. El algoritmo necesario para resolver (P_0) y recuperar la señal buscada podría requerir demasiado tiempo.
2. Si las medidas se encuentran afectadas por ruido, el desempeño del algoritmo de recuperación utilizado podría resultar muy deteriorado.

Como se verá más adelante, para superar estas dos limitantes, y poder recuperar la señal con un algoritmo eficiente y “estable” frente al ruido, será necesario realizar una cantidad mínima de medidas, superior a $m = 2k$, dada por:

$$m = Ck \log \left(e \frac{n}{k} \right).$$

2.2. Propiedad de Isometría Restringida (RIP)

Otra forma de analizar la unicidad de soluciones es mediante la “Propiedad de Isometría Restringida” (RIP, por sus siglas en inglés), introducida por Candès y Tao en [18]:

Definición 3. *([95][Definición 1.3]) $\Phi \in \mathbb{R}^{m \times n}$ cumple la propiedad RIP de orden k si existe $\delta_k \in [0, 1]$ tal que:*

$$(1 - \delta_k) \|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_k) \|x\|_2^2, \quad \forall x \in \Sigma_k.$$

Es decir que Φ se comporta aproximadamente como una isometría al ser restringida al conjunto de señales k -esparsas Σ_k . Al menor δ_k que verifica la propiedad RIP de orden k se le denomina constante de la isometría restringida (RIC). El siguiente teorema brinda una condición suficiente para la k -unicidad de soluciones de CS mediante la propiedad RIP.

Teorema 4. ([19][Teorema 1.1]) *Supongamos que Φ satisface la propiedad RIP de orden $2k$, con constante $\delta_{2k} < 1$. Entonces se tiene k -unicidad.*

Se puede realizar una interpretación geométrica del Teorema 4, basada en que la propiedad RIP de orden $2k$ es equivalente a las siguientes desigualdades en Σ_k :

$$(1 - \delta_{2k}) \|x_2 - x_1\|_2^2 \leq \|\Phi x_2 - \Phi x_1\|_2^2 \leq (1 + \delta_{2k}) \|x_2 - x_1\|_2^2, \forall x_1, x_2 \in \Sigma_k.$$

Es decir: cuanto más pequeño sea $\delta_{2k} \in [0, 1)$, tanto más se preservará la distancia entre dos vectores k -esparsos al ser transformados por Φ (Figura 2.1). En particular, como $1 - \delta_{2k} > 0$, es posible diferenciar dos vectores k -esparsos conociendo sus transformados por Φ , pues estos últimos serán distintos. De ahí la unicidad.

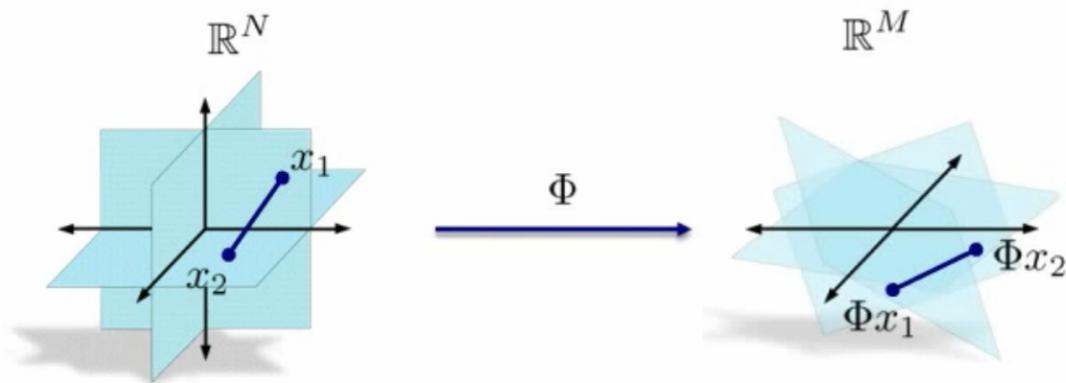


Figura 2.1: Si Φ cumple la propiedad RIP de orden $2k$, con constante $\delta_{2k} < 1$, se tiene: para todo par de vectores k -esparsos distintos, sus transformados por Φ serán distintos, garantizando k -unicidad en (P_0) .

Observación 1. *La cota dada en el Teorema 4 es justa. Esto es: existe Φ con $\delta_{2k} = 1$ y tal que no cumple k -unicidad [19]. Sin embargo, dada una matriz Φ , no es necesario que esta cumpla la propiedad RIP con $\delta_{2k} < 1$ para tener k -unicidad. Consideremos por ejemplo la matriz:*

$$\Phi = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Veamos primero que para el problema CS asociado a Φ se tiene 1-unicidad. El núcleo de Φ es:

$$N(\Phi) = \{(\alpha, -3\alpha, \alpha), \alpha \in \mathbb{R}\}.$$

Es decir que este no contiene a ningún vector 2-esparso: $N(\Phi) \cap \Sigma_2 = \{\vec{0}\}$. Por el Teorema 1 se concluye que se tiene 1-unicidad. Se puede llegar al mismo resultado usando el Teorema 2. En efecto,

a partir de la expresión hallada para el núcleo de Φ , se tiene:

$$\text{spark}(\Phi) = \min\{j \in \{1, 2, 3\} / N(\Phi) \cap \Sigma_j \neq \{\vec{0}\}\} = 3 > 2.$$

Se verá ahora que, a pesar de tener 1-unicidad, Φ no cumple con la propiedad RIP(2), para ningún $\delta_2 < 1$. Tomemos por ejemplo $\hat{x} = (1, 0, 0) \in \Sigma_2$. Se tiene:

$$\|\Phi\hat{x}\|_2^2 = \|(2, 1)\|_2^2 = 5 = 5\|(1, 0, 0)\|_2^2 = 5\|\hat{x}\|_2^2 > (1 + \delta_2)\|\hat{x}\|_2^2, \forall \delta_2 < 1.$$

Como se verá mas adelante, a pesar de no ser necesaria para tener k -unicidad, y de que su cálculo es en general difícil, la propiedad RIP resulta muy útil para garantizar el buen desempeño de algoritmos que buscan resolver (CS) de forma eficiente; especialmente en el caso en que las medidas están afectadas por ruido o las señales no son exactamente k -esparsas.

2.3. Coherencia mutua de Φ

Si bien la propiedad RIP y el spark permiten garantizar la unicidad de soluciones de CS, el cálculo de estos es difícil en general, encontrándose dentro de los problemas NP-difícil [67]. Una propiedad cuyo cálculo resulta más sencillo es la coherencia mutua de Φ .

Definición 4. ([35][Definición 1.5]) Sea $\Phi \in \mathbb{R}^{m \times n}$, con columnas ϕ^i no nulas. Se define la coherencia mutua de Φ como:

$$\mu(\Phi) := \max_{1 \leq i, j \leq n, i \neq j} \frac{|\langle \phi^i, \phi^j \rangle|}{\|\phi^i\|_2 \|\phi^j\|_2}.$$

Donde $\langle u, v \rangle := u^T v$ es el producto interno usual. Si $m = n$, es posible obtener matrices con $\mu(\Phi) = 0$, tomando Φ ortogonal. En el caso de interés, en el cual $m < n$, se tiene [35][Sección 1.4]:

$$0 < \sqrt{\frac{n-m}{m(n-1)}} \leq \mu(\Phi) \leq 1.$$

La cota inferior se conoce como “cota de Welch” y es alcanzada por ejemplo por las matrices denominadas “Grassmannian frames”, las cuales tienen además spark máximo: $\text{spark}(\Phi) = m + 1$ [13][Sección 2]. En el caso general la coherencia y el spark se encuentran vinculados por la siguiente desigualdad [35][Lema 1.4]:

$$\text{spark}(\Phi) \geq 1 + \frac{1}{\mu(\Phi)}.$$

Recordando que $\text{spark}(\Phi) > 2k$ garantiza k -unicidad, se tiene la siguiente condición suficiente:

Teorema 5. ([35][Teorema 1.7]) Si $k < \frac{1}{2} \left(1 + \frac{1}{\mu(\Phi)}\right)$, se tiene k -unicidad.

Es deseable entonces que la coherencia mutua de Φ sea la menor posible, y así garantizar k -unicidad para el mayor rango posible de esparsidad k . Sin embargo, aún en el caso más favorable en que Φ tiene coherencia mutua mínima, el resultado anterior es más débil que los obtenidos mediante el spark. Para ver esto se considera el siguiente ejemplo:

Para las matrices “Grassmannian frames”, cuya coherencia mutua es mínima: $\mu(\Phi) = \frac{1}{\sqrt{m}}$, el Teorema 5 anterior permite garantizar k -unicidad para todo $k \leq \frac{\sqrt{m}}{2}$. Por otro lado, estas matrices tienen $\text{spark}(\Phi) = m + 1$, por lo que el Teorema 2 permite aumentar esta cota y garantizar k -unicidad para todo $k \leq \frac{m}{2}$.

En general, si bien la coherencia mutua es fácil de calcular, los resultados teóricos obtenidos con esta resultan más débiles que los basados en el spark o la propiedad RIP. Por este motivo, en lo que sigue no se considerarán los resultados basados en la coherencia mutua.

Capítulo 3

Minimización ℓ_1

En este capítulo se presentan algunos resultados teóricos sobre el problema convexo (P_1) , denominado Basis Pursuit, y sobre su equivalencia con el problema CS, dado por (P_0) . La equivalencia entre estos, que no siempre puede ser garantizada, resulta muy útil para resolver CS de forma eficiente. Se considera en particular el caso de mayor interés práctico en que las medidas están afectadas por ruido.

3.1. Uso de la norma ℓ_1

Antes de estudiar el problema (P_1) , conviene analizar las razones por las cuales este resulta apropiado como alternativa a (P_0) . Podría parecer más natural reemplazar la pseudo-norma ℓ_0 por la norma usual ℓ_2 , y resolver:

$$\begin{aligned} \operatorname{argmín} \quad & \|x\|_2^2. & (P_2) \\ \Phi x &= b \\ x &\in \mathbb{R}^n \end{aligned}$$

El Problema (P_2) es fácil de resolver en general. En particular si Φ es de rango máximo m , la solución es única y viene dada por:

$$\hat{x} = \Phi^+ b = \Phi^T (\Phi \Phi^T)^{-1} b.$$

Sin embargo, las soluciones de (P_2) no suelen ser las más esparsas del conjunto de soluciones de $\Phi x = b$. Es decir que por lo general no coinciden con la solución de (P_0) . La razón por la cual ocurre esto se puede comprender de forma intuitiva, analizando la geometría de la bola generada por cada norma. En la Figura 3.1 se representa el conjunto de soluciones de un sistema indeterminado $\Phi x = b$, dado en este caso por una recta (en azul), junto con las respectivas soluciones de (P_2) y (P_1) , denotadas x_{LS} y x_{BP} respectivamente. Se representan además las bolas obtenidas con cada norma, centradas en el origen.

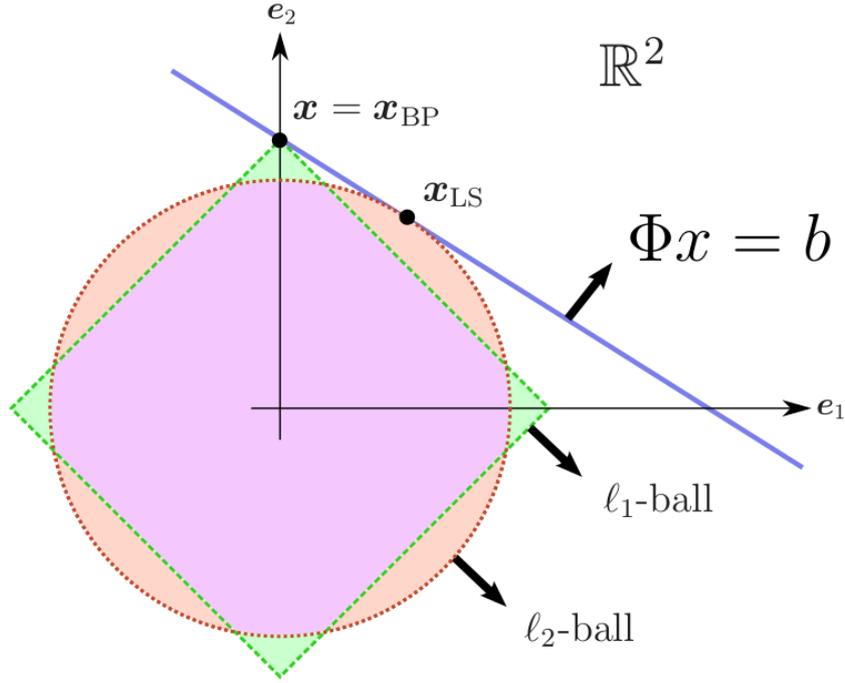


Figura 3.1: La solución x_{BP} , que pertenece al conjunto de soluciones de $\Phi x = b$ (recta azul), a la vez que minimiza la norma ℓ_1 , coincide con x , una de las dos soluciones más esparsas del sistema lineal. Por otro lado, la solución x_{LS} , obtenida minimizando la norma ℓ_2 , es menos esparsa. (Imagen tomada de [44]).

Las soluciones más esparsas de $\Phi x = b$, y por lo tanto las soluciones de (P_0) , son las dos que pertenecen a los ejes coordenados. El problema (P_1) tiene como solución única a una de ellas, denotada en este caso por x . Por otro lado, la solución única de (P_2) es menos esparsa. Esto se debe a la forma geométrica de la bola de cada norma, siendo la de ℓ_1 más favorable a la esparcidad.

Observación 2. En el ejemplo anterior (P_0) tiene más de una solución. Como ya se ha mencionado, en la práctica nos interesa que este tenga una única solución, para poder recuperar la señal de forma unívoca. De todas formas el ejemplo es útil para comprender intuitivamente la ventaja de la norma ℓ_1 sobre la norma ℓ_2 .

Lo anterior indica que es preferible el uso de ℓ_1 sobre ℓ_2 para buscar soluciones esparsas. Veamos ahora qué ventaja presenta el uso de ℓ_1 frente a otras funciones. Es posible generalizar (P_1) y (P_2) considerando el siguiente problema (P_q) :

$$\begin{aligned} \operatorname{argmín} \quad & \|x\|_q^q. & (P_q) \\ \Phi x = & b \\ x \in & \mathbb{R}^n \end{aligned}$$

Donde, para cada $q > 0$, se define la función ℓ_q como:

$$\ell_q(x) = \|x\|_q := \left(\sum_{i=1}^n |x_i|^q \right)^{\frac{1}{q}}.$$

Las funciones ℓ_q constituyen una norma para todo $q \geq 1$; mientras que para $0 < q < 1$ son una cuasi-norma (quasi-norm).¹ En la Figura 3.2 se representan estas funciones para algunos valores de $q > 0$.

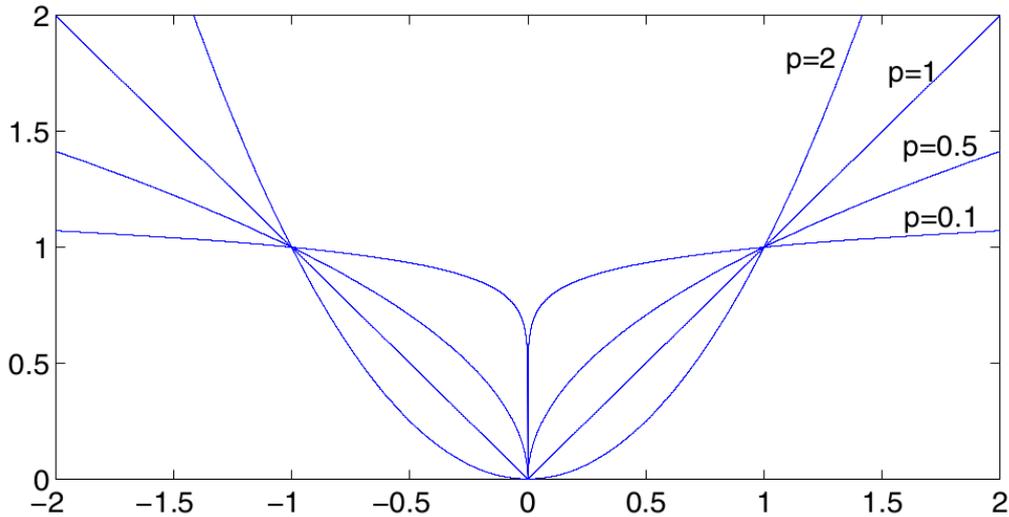


Figura 3.2: Funciones ℓ_q para distintos valores de q en el caso unidimensional ($n = 1$). A medida que q tiende a cero, ℓ_q tiende a la pseudo-norma ℓ_0 . (Imagen tomada de [13]).

Como se puede ver, ℓ_q se aproxima a ℓ_0 cuando q tiende a cero:

$$\lim_{q \rightarrow 0^+} \|x\|_q^q = \|x\|_0.$$

Esto hace que el problema (P_q) resulte particularmente atractivo para $q \simeq 0$. Sin embargo, cuando $0 < q < 1$, las funciones ℓ_q no son convexas, y el problema asociado (P_q) resulta NP-difícil [38][Sección 4.1]. La norma ℓ_1 constituye entonces la función convexa más “cercana” a ℓ_0 de todas las del tipo ℓ_q , $q > 0$.

3.2. Problema ℓ_1 sin ruido

Se considerará entonces el uso de la norma ℓ_1 para buscar soluciones de CS. Se analiza inicialmente la equivalencia entre (P_0) y (P_1) para el caso en que las medidas b no están afectadas por ruido. El problema (P_1) está dado por:

$$\begin{aligned} \operatorname{argmín} \quad & \sum_{i=1}^n |x_i|. & (P_1) \\ \Phi x = & b \\ x \in & \mathbb{R}^n \end{aligned}$$

¹Una cuasi-norma cumple todas las propiedades de una norma, excepto por la desigualdad triangular.

3.2.1. Equivalencia entre (P_1) y (P_0)

El siguiente teorema brinda condiciones suficientes para recuperar cualquier solución de (P_0) mediante la resolución de (P_1) .

Teorema 6. ([35][Teorema 1.8]) *Supongamos que Φ satisface la propiedad RIP de orden $2k$, con constante $\delta_{2k} < \sqrt{2} - 1$. Supongamos además que las medidas utilizadas son: $b = \Phi x^*$. Entonces el error entre cualquier solución \hat{x} de (P_1) y la (única) solución x^* de (P_0) verifica:*

$$\|\hat{x} - x^*\|_2 \leq C_0 \frac{\sigma_k(x^*)_1}{\sqrt{k}}, \quad C_0 = 2 \frac{1 - (1 - \sqrt{2})\delta_{2k}}{1 - (1 + \sqrt{2})\delta_{2k}};$$

donde $\sigma_k(x^*)_1$ es la distancia de x^* al conjunto de señales k -esparsas, tal como fuera definida en el capítulo de introducción. En particular el error es nulo cuando x^* es k -esparsa ($\sigma_k(x^*)_1 = 0$), por lo que en ese caso la minimización ℓ_1 permite recuperar x^* .

Observación 3. *La cota $\delta_{2k} < \sqrt{2} - 1 \simeq 0.414$, utilizada en el Teorema 6, no es óptima, y por lo tanto puede ser mejorada. Por ejemplo recientemente se ha logrado extender esta cota a [39][Teorema 1]:*

$$\delta_{2k} < \frac{4}{\sqrt{41}} \simeq 0.625.$$

Por otro lado en [27][Teorema 1] se prueba que la cota no puede superar el valor $\frac{1}{\sqrt{2}} \simeq 0.707$. Esto es: $\forall \epsilon > 0$, existe una matriz Φ , con $\delta_{2k} \leq \frac{1}{\sqrt{2}} + \epsilon$, y tal que no es posible recuperar alguna solución k -esparsa mediante minimización ℓ_1 . Por otro lado, existen resultados similares al del Teorema 6, pero que utilizan una cota sobre δ_k , en lugar de δ_{2k} . Por ejemplo en [14][Teorema 3.1] se prueba que $\delta_k < \frac{1}{3}$ es suficiente, para todo $k \geq 2$. En este caso la cota es óptima, pues existe Φ con $\delta_k = \frac{1}{3}$, $k \geq 2$, y tal que algunas soluciones k -esparsas no pueden ser recuperadas mediante minimización ℓ_1 [14][Teorema 3.2]. El caso $k = 1$ queda excluido de los teoremas basados en una cota sobre δ_k , pues es posible construir Φ con $\delta_1 = 0$, y tal que no verifica 1-unicidad. Para ver esto basta tomar $\Phi \in \mathbb{R}^{m \times n}$, con $n = m + 1$ y tal que:

$$\Phi x := (x_1 - x_2, x_3, \dots, x_n) \in \mathbb{R}^m, \quad \forall x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n.$$

Se ve fácilmente que Φ cumple $RIP(1)$ con $\delta_1 = 0$: $\|\Phi x\|_2^2 = \|x\|_2^2, \forall x \in \Sigma_1$. Sin embargo no se tiene 1-unicidad pues: $\Phi e_1 = \Phi(-e_2)$; siendo $e_i \in \Sigma_1$ y tal que $e_i = 1$. En particular esto implica que no es posible garantizar la recuperación de las soluciones e_1 o $-e_2$. Mas allá de estas observaciones, se utiliza el enunciado dado en el Teorema 6 por ser la versión clásica de este tipo de resultados, introducida por Candès en [19].

Cabe preguntarse ahora cuándo es posible garantizar que Φ cumple la propiedad RIP de orden $2k$. El siguiente teorema brinda una condición necesaria para ello.

Teorema 7. ([26][Teorema 3.5]) *Si Φ satisface la propiedad RIP de orden $k \leq \frac{n}{2}$, con constante*

$\delta \in (0, 1)$, se debe cumplir:

$$m \geq C_\delta k \log \left(\frac{n}{k} \right), \quad C_\delta \simeq \frac{0.18}{\log \left(\sqrt{\frac{1+\delta}{1-\delta}} + 1 \right)} < 1.$$

Los dos últimos teoremas sugieren que, en el mejor caso, y trabajando en base a la propiedad RIP($2k$), la resolución de (P_1) permite recuperar cualquier señal k -esparsa disponiendo de una cantidad de medidas del orden de:

$$m = O \left(k \log \left(\frac{n}{k} \right) \right).$$

Sin embargo, no se conocen matrices determinísticas para las cuales $m = O \left(k \log \left(\frac{n}{k} \right) \right)$ sea suficiente para cumplir la propiedad RIP de orden $2k$, con constante $\delta_{2k} < \sqrt{2} - 1$. Los mejores resultados utilizan al menos $m \simeq O(k^2)$ medidas [11]. De todas formas, como lo muestra el siguiente teorema, si se consideran matrices aleatorias, es posible alcanzar la cota inferior de medidas y aún así cumplir la propiedad RIP deseada con alta probabilidad.

3.2.2. Medidas con Φ aleatoria

Teorema 8. ([38][Teorema 9.27]) Sea $\Phi \in \mathbb{R}^{m \times n}$, con $m < n$, una matriz aleatoria con entradas Gaussianas estándar:

$$\Phi_{ij} \sim N(0, 1), \text{ independientes.}$$

Dados $\epsilon, \gamma \in (0, 1)$, si la cantidad m de medidas verifica:

$$m \geq \frac{2}{\gamma^2} \left(k \ln \left(e \frac{n}{k} \right) + \log \left(\frac{2}{\epsilon} \right) \right),$$

entonces, con probabilidad al menos $1 - \epsilon$, la matriz normalizada $\bar{\Phi} = \frac{1}{\sqrt{m}}\Phi$ cumple la propiedad RIP con constante:

$$\delta_k \leq 2c\gamma + c^2\gamma^2, \quad c = 1 + \frac{1}{\sqrt{2 \ln \left(e \frac{n}{k} \right)}}.$$

El resultado también es válido cuando la distribución de los Φ_{ij} es Bernoulli, con valores equiprobables ± 1 . De hecho el resultado se puede extender a cualquier distribución de tipo “sub-Gaussiana”, con media nula y varianza unidad [38][Teorema 9.2].

Observación 4. Dado que se utiliza $\Phi_{ij} \sim N(0, 1)$, las n columnas de $\bar{\Phi} = \frac{1}{\sqrt{m}}\Phi$ tendrán norma 2 unidad en valor esperado. Es decir que para cada columna $\bar{\Phi}^j$, $j = 1, \dots, n$, se tiene:

$$\mathbb{E}(\|\bar{\Phi}^j\|_2^2) = \mathbb{E} \left(\sum_{i=1}^m \left(\frac{\Phi_{ij}}{\sqrt{m}} \right)^2 \right) = \frac{1}{m} \sum_{i=1}^m \mathbb{E}(\Phi_{ij}^2) = \frac{1}{m} \sum_{i=1}^m \mathbb{V}(\Phi_{ij}) = 1.$$

Por lo tanto las columnas de $\bar{\Phi}$ ya se encuentran normalizadas (en valor esperado).

A partir del Teorema 8 se pueden obtener ejemplos cuantitativos del ahorro en medidas que puede significar el problema CS, respecto a las n medidas del proceso usual de adquisición de señales. Si se

asume un nivel de esparcidad de al menos $\frac{n}{k} \geq 2e^3 \simeq 40.2$, se tiene $c_{2k} \leq 1 + \frac{1}{\sqrt{8}}$. Tomando $\gamma_{2k} = \frac{1}{5}$, se obtiene:

$$\delta_{2k} \leq 2 \left(\frac{c_{2k}}{5} \right) + \left(\frac{c_{2k}}{5} \right)^2 \simeq 0.615 < \frac{4}{\sqrt{41}}.$$

Usando entonces el Teorema 8, junto con el Teorema 6 y las observaciones respecto a la posibilidad de extender la cota de suficiencia sobre δ_{2k} , se puede garantizar que: si

$$m \geq 50 \left(2k \ln \left(e \frac{n}{2k} \right) + \log \left(\frac{2}{\epsilon} \right) \right),$$

Φ cumple RIP($2k$), con $\delta_{2k} < \frac{4}{\sqrt{41}}$, y por lo tanto es posible recuperar cualquier vector k -esparso mediante minimización ℓ_1 , con probabilidad al menos $1 - \epsilon$. En la Figura 3.3 se muestra la cota inferior de medidas obtenida de esta forma, para distintos valores de n y con probabilidad al menos 0.99 ($\epsilon = 0.01$). Las medidas m se muestran en porcentaje respecto de n : $\frac{m}{n} \times 100$. Por ejemplo para $n = 1 \times 10^6$, la minimización ℓ_1 permite recuperar cualquier señal k esparsa, con una cantidad de medidas $m < n$, para todo $k \leq 1500$. De esta forma el problema CS permite adquirir la misma señal $x^* \in \mathbb{R}^n$ pero con menor cantidad de medidas, en relación a las n medidas del proceso usual.

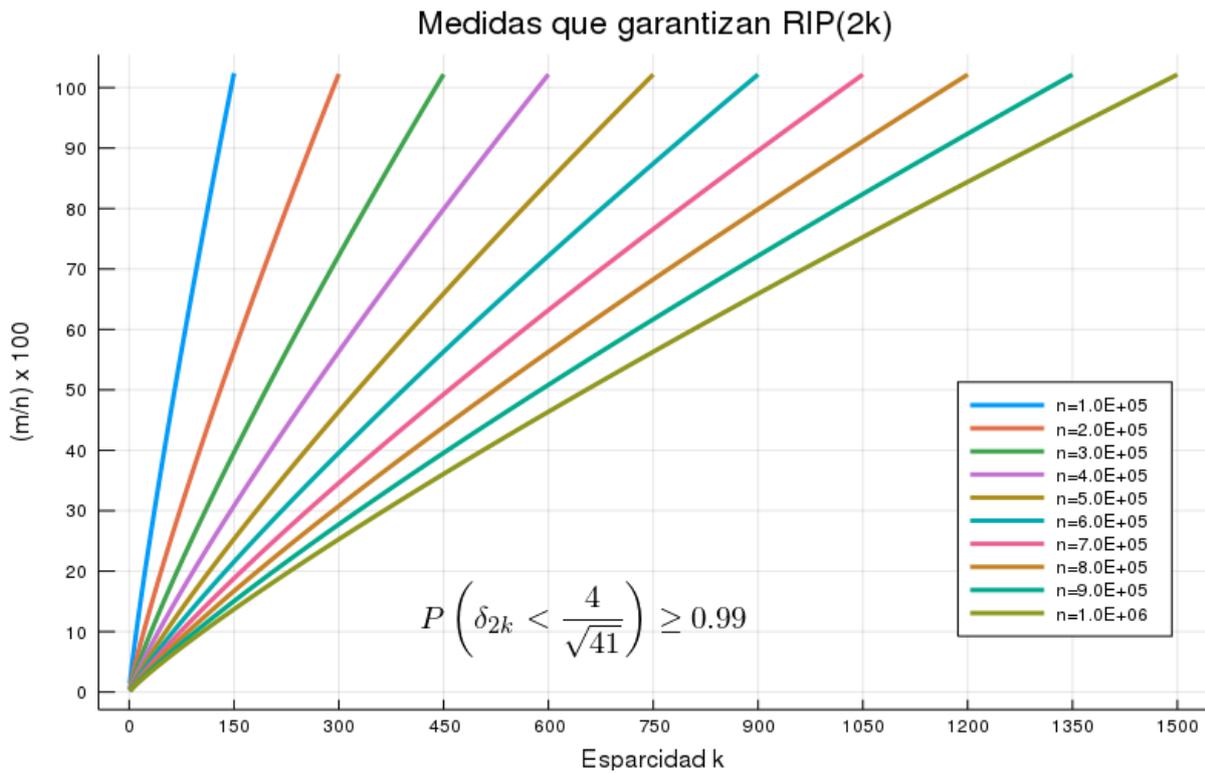


Figura 3.3: Cota inferior de medidas $m = 50 \left(2k \ln \left(e \frac{n}{2k} \right) + \log \left(\frac{2}{\epsilon} \right) \right)$ que garantizan $\delta_{2k} < \frac{4}{\sqrt{41}}$, con probabilidad al menos 0.99 ($\epsilon = 0.01$). Se consideran valores de $n \in [100 \times 10^3, 1 \times 10^6]$.

3.2.3. Fenómeno de transición de fase

Conviene recordar que la cota inferior de medidas obtenida mediante la propiedad RIP, y representada mediante un ejemplo en la Figura 3.3, no es óptima. Donoho y Tanner, utilizando geometría de politopos, logran cotas ajustadas, aunque no siempre se pueden hallar de forma explícita. Para

esto definen las variables $\delta = \frac{m}{n}$, denominada factor de submuestreo y $\rho = \frac{k}{m}$, factor de esparsidad. Prueban entonces que existe una función umbral $\rho(\delta)$ tal que:

$$\lim_{n \rightarrow \infty} P(\text{recuperar cualquier } x^* \in \Sigma_k \text{ con } (P_1)) = \begin{cases} 1 & \text{si } \rho < \rho(\delta) \\ 0 & \text{si } \rho > \rho(\delta) \end{cases}.$$

Es decir que se tiene un comportamiento de tipo transición de fase. Para un n fijo, en lugar de una curva, existirá una “región” de transición, cuyo ancho tiende a cero a medida que $n \rightarrow \infty$. En la Figura 3.4 se muestra en negro la curva de transición de fase asintótica ($n \rightarrow \infty$), asociada a una matriz de medida Φ de tipo Fourier aleatoria (ver Sección 3.4). Para el mismo tipo de problema, pero con $n = 1600$ fijo, se muestran: las curvas de umbral de 90 % de probabilidad de éxito (azul), de 50 % (verde) y de 10 % (roja), obtenidas mediante simulaciones numéricas. Es interesante observar que, aún para este valor relativamente pequeño de n , la región de transición es muy estrecha.

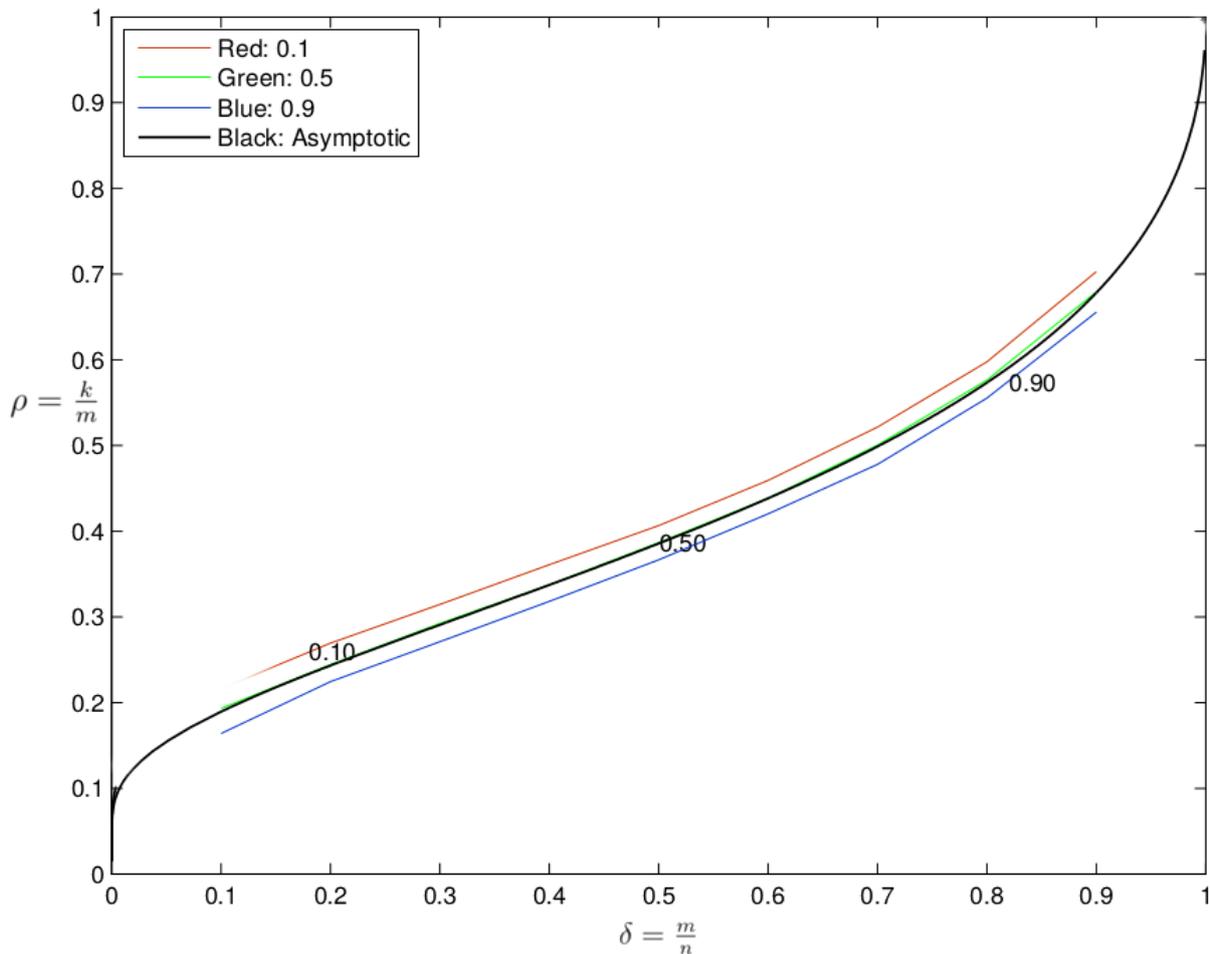


Figura 3.4: Curva de transición de fase asintótica ($n \rightarrow \infty$), asociada a una matriz de Fourier aleatoria. Para igual matriz pero con $n = 1600$ fijo, se muestran: las curvas de umbral de 90 % de probabilidad de éxito (azul), de 50 % (verde) y de 10 % (roja), obtenidas mediante simulaciones numéricas (Imagen tomada de [32]).

3.2.4. RIP y cambio de base

Como fuera mencionado en el capítulo de introducción, por lo general las señales de interés no son esparsas en la base canónica, sino que deben ser expresadas en una base específica para obtener una representación esparsa: $\alpha = \Psi x$. En este caso se busca estimar $\alpha \in \mathbb{R}^n$, por lo que el problema CS tiene matriz $\Phi\Psi^{-1}$. Surge entonces la siguiente dificultad: dada Φ que cumple la propiedad RIP con constante δ_{2k} pequeña, es posible elegir Ψ de forma que la constante RIP de $\Phi\Psi^{-1}$ sea arbitrariamente grande [74]. Es decir que la matriz de cambio de base Ψ podría empeorar el desempeño de la recuperación ℓ_1 , aún cuando Φ cumpla la propiedad RIP con δ_{2k} adecuada. Sin embargo, en el caso en que la matriz de medida Φ es aleatoria, el siguiente teorema garantiza que la constante de la propiedad RIP es preservada por cualquier transformación ortogonal Ψ con alta probabilidad, y siempre que la cantidad de medidas m sea suficientemente grande.

Teorema 9. ([38][Teorema 9.15]) *Sea $\Psi \in \mathbb{R}^{n \times n}$ una matriz ortogonal fija. Supongamos que la matriz de medida $\Phi \in \mathbb{R}^{m \times n}$ es aleatoria, con entradas independientes, de distribución Gaussiana estándar. Entonces, dados $\epsilon, \delta \in (0, 1)$, existe $C > 0$ tal que: si la cantidad de medidas m verifica:*

$$m \geq \frac{C}{\delta^2} \left(k \left(9 + 2 \ln \left(\frac{n}{k} \right) \right) + 2 \ln \left(\frac{2}{\epsilon} \right) \right),$$

la matriz normalizada $\frac{1}{\sqrt{m}}\Phi\Psi^{-1}$ cumple la propiedad RIP con constante $\delta_k < \delta$, con probabilidad al menos $1 - \epsilon$.

En el caso general en que Φ no es aleatoria, existen propiedades alternativas a RIP que se preservan por transformaciones ortogonales. Una de ellas es la denominada “Propiedad del Núcleo” (Null Space Property - NSP) [74], que no será utilizada en este trabajo.

3.3. Problema ℓ_1 con ruido

Se considera ahora el problema más realista (P_0^η), en el cual las medidas están afectadas por ruido z aditivo: $b = \Phi x^* + z$. Como ya fue dicho, al igual que (P_0), el problema (P_0^η) es de tipo NP-difícil, para cualquier nivel de ruido $\eta \geq 0$. Se trabajará entonces con el problema alternativo que se obtiene al reemplazar la pseudo-norma ℓ_0 por la norma ℓ_1 :

$$\begin{aligned} & \text{mín} && \|x\|_1. && (P_1^\eta) \\ & \|\Phi x - b\|_2 \leq \eta && && \\ & x \in \mathbb{R}^n && && \end{aligned}$$

Este es una generalización de (P_1) y se conoce como Basis Pursuit Denoising (BPD). En este caso en general no será posible, ni siquiera en forma teórica, recuperar x^* de forma exacta; por lo que nos limitaremos a buscar una solución de (P_1^η) que sea cercana a x^* .

Observación 5. *Es interesante notar que un problema CS sin ruido, y donde se busca una solución $x^* \in \Sigma_k$, se puede expresar en forma equivalente como un problema con ruido y solución buscada*

l -esparsa, $l \leq k$. Es decir que se introduce ruido pero se obtiene como ventaja una mayor esparcidad. Por ejemplo, partiendo de un modelo sin ruido, y utilizando únicamente la primer coordenada (no nula) de $x^* \in \Sigma_k$ para convertirla en ruido, se tiene:

$$b = \Phi x^* + \vec{0} = \sum_{j=1}^n \Phi^j x_j^* = \sum_{j=2}^n \Phi^j x_j^* + \Phi^1 x_1^* = \Phi \hat{x}^* + z.$$

Donde:

$$\hat{x}^* = [0, x_2^*, \dots, x_n^*] \in \Sigma_{k-1}, \quad z = \Phi^1 x_1^* \in \mathbb{R}^m, \quad z \neq \vec{0}.$$

De forma similar se puede convertir un problema CS con ruido en un problema sin ruido, pero con la desventaja de buscar una solución menos esparsa. Es decir que es posible intercambiar ruido por esparcidad. Esto permite tratar de forma equivalente el ruido z y la no k -esparcidad de una señal x^* , dada por la distancia al conjunto Σ_k : $\sigma_k(x^*)$.

3.3.1. Caso con oráculo

Antes de presentar los resultados teóricos vinculados al desempeño bajo ruido, y sólo a modo de referencia, conviene estudiar el caso ideal, o con oráculo, en que se busca resolver (P_0^η) conociendo el soporte S de la solución buscada x^* . En este caso sólo resta recuperar los k valores no nulos de x^* . Esto se puede hacer resolviendo el problema de mínimos cuadrados asociado a la matriz $\Phi_S \in \mathbb{R}^{m \times k}$, formada por las columnas S de Φ :

$$\hat{x}_S \in \underset{x \in \mathbb{R}^k}{\operatorname{argmín}} \|\Phi_S x - b\|_2^2.$$

Si Φ_S es de rango máximo k , la solución de mínimos cuadrados es única y viene dada mediante la pseudo inversa Φ_S^+ [35][Sección 1.5]:

$$\hat{x}_S = \Phi_S^+ b = (\Phi_S^T \Phi_S)^{-1} \Phi_S^T b.$$

Usando que $b = \Phi x^* + z$, se obtiene entonces el error ideal en función del ruido:

$$\|\hat{x} - x^*\|_2 = \|(\Phi_S^T \Phi_S)^{-1} \Phi_S^T (\Phi x^* + z) - x^*\|_2 = \|(\Phi_S^T \Phi_S)^{-1} \Phi_S^T z\|_2 = \|\Phi_S^+ z\|_2, \quad \forall z \in \mathbb{R}^m.$$

Esto permite obtener una cota superior del error ideal, en términos del mayor valor singular de Φ_S^+ , siendo esta una cota óptima:

$$\max_{z \neq \vec{0}} \frac{\|\Phi_S^+ z\|_2}{\|z\|_2} = \sigma_{\max}(\Phi_S^+) \Rightarrow \|\hat{x} - x^*\|_2 = \|\Phi_S^+ z\|_2 \leq \sigma_{\max}(\Phi_S^+) \|z\|_2, \quad \forall z \in \mathbb{R}^m.$$

Por lo tanto, aún en el caso con oráculo, en que se conoce el soporte de x^* , lo mejor que se puede decir de una solución \hat{x} de (P_0^η) es que su distancia a la señal buscada x^* no supera el nivel de ruido $\|z\|_2$ en más de σ_{\max} .

3.3.2. Estabilidad respecto al ruido

El razonamiento anterior motiva la siguiente definición de estabilidad, aplicable no sólo al algoritmo de minimización ℓ_1 , sino a cualquier algoritmo de recuperación Δ .

Definición 5. ([35][Definición 1.4]) *Se considera una matriz de medida $\Phi \in \mathbb{R}^{m \times n}$ y un algoritmo de recuperación $\Delta : \mathbb{R}^m \rightarrow \mathbb{R}^n$, aplicado a resolver (P_0^η) . El par (Φ, Δ) se denomina C -estable si:*

$$\|\Delta(\Phi x^* + z) - x^*\|_2 \leq C \|z\|_2, \quad \forall z \in \mathbb{R}^m, \forall x^* \in \Sigma_k.$$

Es decir: el error entre la señal recuperada por el algoritmo: $\Delta(\Phi x^* + z)$, y la señal buscada x^* , no supera el nivel del ruido en más de C . Como se vio en el Capítulo 2, en el caso sin ruido es posible recuperar cualquier señal k -esparsa con tan sólo $m = 2k$ medidas; utilizando por ejemplo una matriz de tipo Vandermonde. Sin embargo, como lo muestra el siguiente teorema, esta cantidad de medidas es insuficiente si se desea que el proceso de recuperación sea estable respecto al ruido.

Teorema 10. *Se considera una matriz de medida $\Phi \in \mathbb{R}^{m \times n}$ y un algoritmo de recuperación $\Delta : \mathbb{R}^m \rightarrow \mathbb{R}^n$. Si (Φ, Δ) es C -estable para señales k -esparsas, entonces existe una constante $C > 0$ tal que:*

$$m \geq Ck \ln \left(e \frac{n}{k} \right).$$

Los siguientes dos teoremas, junto con el Teorema 8, permiten concluir que el algoritmo de recuperación ℓ_1 es estable respecto al ruido (con alta probabilidad), usando la menor cantidad de medidas posible. Es decir que el algoritmo ℓ_1 es óptimo en ese sentido, y sólo podría ser superado por un algoritmo con menor constante $C > 0$.

3.3.3. Ruido acotado

Si se asume que el ruido es acotado, se tiene la siguiente generalización del Teorema 6.

Teorema 11. ([35][Teorema 1.9]) *Supongamos que Φ satisface la propiedad RIP de orden $2k$, con constante $\delta_{2k} < \sqrt{2} - 1$. Supongamos que las medidas utilizadas son: $b = \Phi x^* + z$, con ruido acotado: $\|z\|_2 \leq \eta$. Entonces cualquier solución \hat{x} de (P_1^η) verifica:*

$$\|\hat{x} - x^*\|_2 \leq C_0 \frac{\sigma_k(x^*)_1}{\sqrt{k}} + C_2 \eta, \quad C_2 = 4 \frac{\sqrt{1 + \delta_{2k}}}{1 - (1 + \sqrt{2})\delta_{2k}}.$$

En particular la componente de error debida al ruido es proporcional al nivel de ruido η , tal como en el caso con oráculo en que se conoce el soporte de x^* . Al igual que para el caso sin ruido, es posible aumentar la cota de suficiencia a $\delta_{2k} < \frac{4}{\sqrt{41}} \simeq 0.625$ [39][Teorema 1]. En la Figura 3.5 se ilustra el resultado de este teorema para el caso de una señal 1-esparsa, en \mathbb{R}^2 .

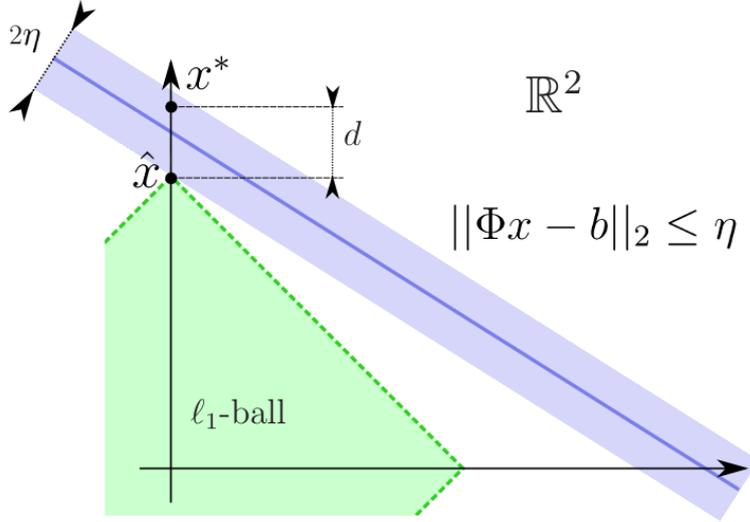


Figura 3.5: El punto \hat{x} , perteneciente al conjunto factible $\|\Phi x - b\|_2 \leq \eta$, y que minimiza la norma ℓ_1 , se encuentra a una distancia de x^* que es proporcional a 2η (Imagen tomada de [44]).

Una desventaja del resultado anterior es que la constante de proporcionalidad del ruido crece indefinidamente al aumentar δ_{2k} :

$$\lim_{\delta_{2k} \rightarrow (\sqrt{2}-1)^-} C_2 = +\infty.$$

De todas formas, para valores pequeños de δ_{2k} , la constante toma valores razonables. Por ejemplo, si $\delta_{2k} = \frac{1}{3}$, se tiene $C_2 \simeq 23.7$.

3.3.4. Ruido Gaussiano

El teorema anterior es válido para cualquier tipo de ruido, siempre que se lo considere acotado. Se puede obtener un resultado similar asumiendo que el ruido es Gaussiano y $x^* \in \Sigma_k$.

Teorema 12. ([35][Corolario 1.1]) *Supongamos que Φ satisface la propiedad RIP de orden $2k$, con constante $\delta_{2k} < \sqrt{2} - 1$. Supongamos que las medidas utilizadas son: $b = \Phi x^* + z$, donde $x^* \in \Sigma_k$ y*

$$z_i \sim N(0, \sigma^2), \text{ independientes.}$$

Entonces existe $c_0 > 0$ tal que: cualquier solución \hat{x} de (P_1^η) , con $\eta = 2\sqrt{m}\sigma$, verifica:

$$\mathbb{P}(\|\hat{x} - x^*\|_2 \leq C_2 \eta) \geq 1 - e^{-c_0 m}.$$

La constante de proporcionalidad C_2 del ruido coincide con la del Teorema 11 anterior. En particular esta crece indefinidamente cuando $\delta_{2k} \rightarrow (\sqrt{2} - 1)^-$. La razón por la cual este resultado se expresa en términos de probabilidad, es que se basa en aplicar el Teorema 11 con la siguiente cota de probabilidad, válida para cualquier vector con coordenadas Gaussianas e independientes $z_i \sim N(0, \sigma^2)$:

$$\mathbb{P}(\|z\|_2 \geq 2\sqrt{m}\sigma) \leq e^{-c_0 m}.$$

Observación 6. *Los dos últimos teoremas tienen como hipótesis la propiedad RIP con $\delta_{2k} < \sqrt{2} - 1$. Como quedó establecido en el Teorema 8, las matrices aleatorias Gaussianas cumplen esta propiedad con alta probabilidad.*

3.4. Matrices aleatorias con estructura

El uso de Φ aleatoria Gaussiana garantiza el buen desempeño del algoritmo de recuperación basado en resolver (P_1^η) . Sin embargo, este tipo de matrices, al no tener una estructura particular, presentan una desventaja práctica. Esto se debe a que los algoritmos utilizados para resolver (P_1^η) suelen realizar varias iteraciones, donde en cada una es necesario multiplicar Φ por algún vector. Para una matriz sin estructura, como el caso de las Gaussianas, este tipo de multiplicación requiere el máximo de operaciones: $O(nm)$. Una alternativa consiste en utilizar matrices aleatorias pero con cierta estructura, como por ejemplo matrices aleatorias de Fourier. Estas se construyen tomando m filas aleatorias de la matriz F asociada a la Transformada Discreta de Fourier (DFT):

$$F = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & r & r^2 & \dots & r^{n-1} \\ 1 & r^2 & (r^2)^2 & \dots & (r^{n-1})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & r^{n-1} & (r^2)^{n-1} & \dots & (r^{n-1})^{n-1} \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad r = e^{-\frac{2\pi i}{n}}.$$

En este caso la multiplicación se puede realizar en $O(n \log(n))$ operaciones, utilizando el algoritmo de la “Transformada Rápida de Fourier” (FFT). La desventaja es que los mejores resultados teóricos indican que es suficiente contar con $m \geq Ck \log^4(n)$ medidas para garantizar RIP con δ_k pequeña (con alta probabilidad) [44][Sección 5].

Capítulo 4

Algoritmos Re-Weighted ℓ_1

En este capítulo se presenta el algoritmo Re-Weighted ℓ_1 ($RW\ell_1$) propuesto por Candès, Wakin y Boyd (CWB) en [20]. Este busca estimar los pesos w_i de forma que las soluciones del problema (P_1W) asociado sean solución de (P_0). Para esto genera una sucesión de pesos w^k , resolviendo en cada paso un problema de tipo Weighted- ℓ_1 . Como se verá en los capítulos 10 y 11 mediante simulaciones, una de las ventajas de $RW\ell_1$ es que, si se fija la cantidad de medidas m , este algoritmo permite en algunos casos recuperar señales que presentan menor esparcidad (mayor k), en relación a la minimización ℓ_1 .

4.1. Algoritmo $RW\ell_1$ de CWB

Los algoritmos $RW\ell_1$ buscan pesos w_i tales que la solución del problema asociado (P_1W) coincida con la solución buscada de (P_0). El problema (P_1W) viene dado por:

$$\begin{aligned} \text{argmín} \quad & \sum_{i=1}^n w_i |x_i|. \\ \Phi x = b \quad & \\ x \in \mathbb{R}^n \quad & \end{aligned} \tag{P_1W}$$

Como se puede ver, los pesos determinan el “costo” de cada coordenada. Por lo tanto, si w_i es grande, la coordenada x_i tenderá a ser pequeña. De igual forma, si el costo w_i es pequeño, x_i podrá ser grande. En el caso ideal en que se conoce una solución x^* de (P_0), se pueden tomar los pesos como:

$$w_i = \begin{cases} +\infty & \text{si } x_i^* = 0 \\ \frac{1}{|x_i^*|} & \text{si } x_i^* \neq 0 \end{cases} .$$

Esto sugiere la siguiente forma de actualizar los pesos, en el caso no ideal en que x^* es desconocida:

$$w_i^{k+1} = \frac{1}{|x_i^{k+1}| + \epsilon}, \quad \forall i = 1, 2, \dots, n.$$

Donde $\epsilon > 0$ se utiliza para evitar dividir por cero y x^{k+1} es alguna estimación de x^* , disponible en

el paso k . Dado que w^k es la estimación de los pesos en dicho paso, parece razonable tomar:

$$x^{k+1} \in \underset{x \in \mathbb{R}^n}{\operatorname{argmín}} \sum_{i=1}^n w_i^k |x_i|. \quad (4.1)$$

Este procedimiento, que denominaremos $\text{RW}\ell_1$ de CWB, es el propuesto en [20] por Candès, Wakin y Boyd. En el Algoritmo 1 se muestra un pseudocódigo del mismo. El algoritmo se implementa de forma que, si no se realizan iteraciones de RW ($\text{maxIter} = 0$), y los pesos iniciales son $w^0 = \vec{1}$, entonces devuelve la solución obtenida por minimización ℓ_1 . Es decir que las iteraciones del algoritmo son literalmente iteraciones de Re-Weighted.

Algorithm 1 Re-Weighted con pesos actualizados según Candès, Wakin y Boyd [20] (sin ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\text{maxIter} \geq 1$, $\epsilon > 0$

```

1:  $x^1 \in \underset{x \in \mathbb{R}^n}{\operatorname{argmín}} \Phi x = b \quad \sum_{i=1}^n w_i^0 |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^0\}$ 
2:
3:  $k = 0$ 
4: while  $k < \text{maxIter}$  do
5:    $w_i^{k+1} = \frac{1}{|x_i^k| + \epsilon}$ 
6:
7:    $k = k + 1$ 
8:
9:    $x^0 = x^k$ 
10:   $x^{k+1} \in \underset{x \in \mathbb{R}^n}{\operatorname{argmín}} \Phi x = b \quad \sum_{i=1}^n w_i^k |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^k\}$ 
11: end while
12: return  $x^{k+1}$ 

```

Intuitivamente, el algoritmo busca actualizar los pesos de forma que el valor funcional mínimo de $(P_1 W)$, se aproxime al valor óptimo $\|x^*\|_0$ de (P_0) :

$$\lim_{k \rightarrow \infty} \|W^k x^k\|_1 = \|x^*\|_0.$$

En efecto, si se asume convergencia de x^k a la solución buscada, y $\epsilon > 0$ es suficientemente pequeño, se tiene:

$$\lim_{k \rightarrow \infty} \|W^k x^k\|_1 = \lim_{k \rightarrow \infty} \sum_{i=1}^n w_i^k |x_i^k| = \lim_{k \rightarrow \infty} \sum_{i=1}^n \left(\frac{1}{|x_i^k| + \epsilon} \right) |x_i^k| = \sum_{i=1}^n \frac{|x_i^*|}{|x_i^*| + \epsilon} \simeq \sum_{x_i^* \neq 0} 1 = \|x^*\|_0.$$

4.2. Marco teórico del $\text{RW}\ell_1$ de CWB

El algoritmo $\text{RW}\ell_1$ descrito se puede ver como un caso particular de una metodología de optimización, conocida como Mayorización-Minimización (MM) [62]. En el caso general, esta metodología

permite obtener un algoritmo para resolver un problema de la siguiente forma, con $C \subset \mathbb{R}^n$ convexo:

$$\begin{aligned} \operatorname{argmín} \quad & g(v). \\ v \in & C \end{aligned} \tag{4.2}$$

El algoritmo obtenido con MM inicia en un punto factible $v^0 \in C$. En el paso k -ésimo se dispone de una estimación v^k de la solución y de una función h mayorizante de g en v^k . Esto es:

$$\begin{cases} h(v) \geq g(v), \forall v \in C \\ h(v^k) = g(v^k) \end{cases} .$$

Luego se actualiza la estimación minimizando el mayorizante, en lugar de g :

$$\begin{aligned} v^{k+1} \in \operatorname{argmín} \quad & h(v). \\ v \in & C \end{aligned} \tag{4.3}$$

En la Figura 4.1 se ilustra la metodología del algoritmo MM, utilizando funciones cuadráticas h como mayorizantes de g .

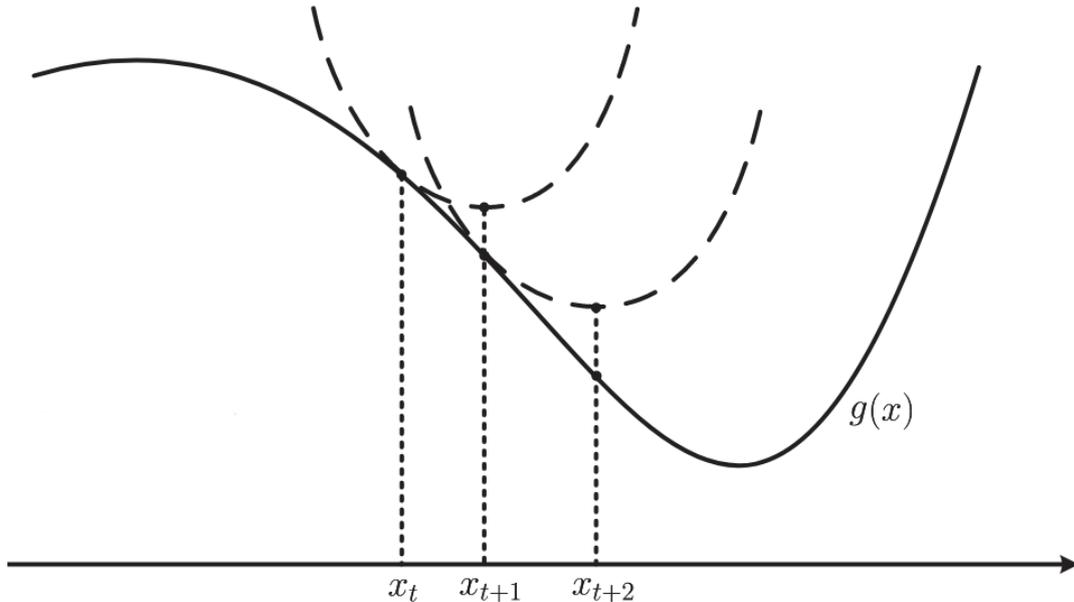


Figura 4.1: Dos pasos del algoritmo MM, utilizado para minimizar una función $g(x)$. Las funciones mayorizantes de cada paso se muestran con líneas punteadas. (Imagen tomada de [62]).

Si la función objetivo g es cóncava, como ocurrirá en nuestro caso de interés, siempre se puede considerar como función mayorizante la linealización de g en torno a v^k . El algoritmo que se obtiene con MM es entonces:

$$\begin{aligned} v^{k+1} \in \operatorname{argmín} \quad & (g(v^k) + \langle \nabla g(v^k), v - v^k \rangle) \\ v \in & C \end{aligned} = \operatorname{argmín} \quad \langle \nabla g(v^k), v \rangle. \tag{4.4}$$

En la Figura 4.2 se muestra un paso del algoritmo MM, utilizado para minimizar una función

cóncava g , en el conjunto convexo $C = [-1, 4]$. En este caso la función tiene mínimo global en $d = (-1, g(-1))$. Como g es cóncava, se utiliza su aproximación lineal como función mayorizante. Si el algoritmo MM se inicia en el punto c , entonces converge al mínimo global de g en un único paso. Sin embargo, si se inicia en el punto a , converge a un mínimo local $b = (4, g(4))$, que no es global. Esto muestra que el algoritmo MM puede converger a un mínimo que no sea global, dependiendo de la condición inicial.

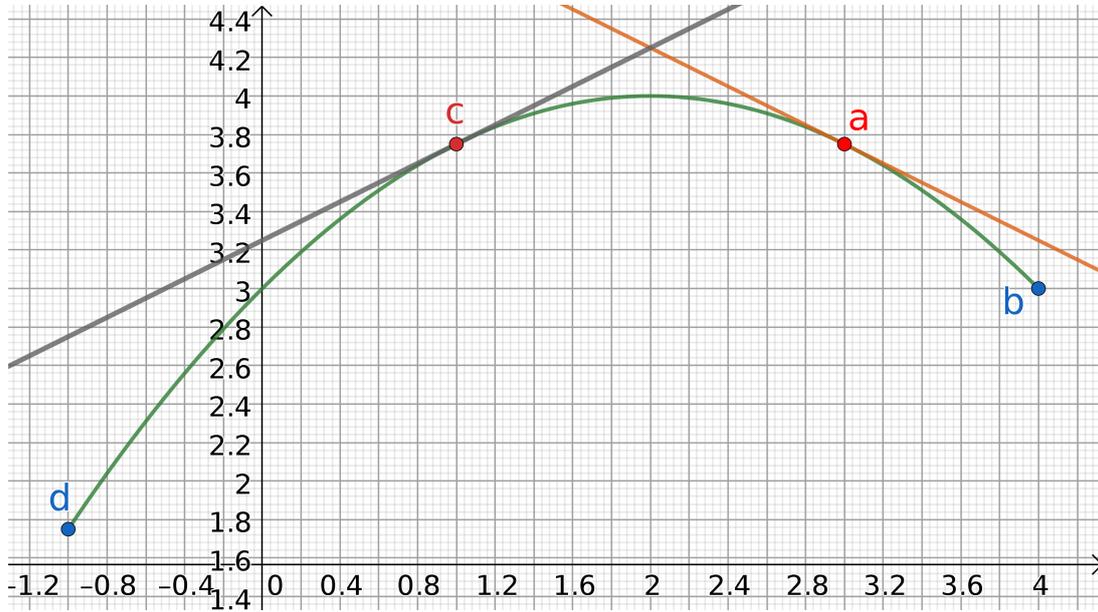


Figura 4.2: Algoritmo MM utilizado para minimizar una función cóncava g (en verde) en el intervalo convexo $C = [-1, 4]$. En cada punto se utiliza la respectiva aproximación lineal de g como función mayorizante (rectas por a y c). Dependiendo de la condición inicial, el algoritmo converge al mínimo global d o a un mínimo local b de g , que no es global.

Se verá ahora cómo se vincula el algoritmo $\text{RW}\ell_1$ de CWB con la metodología MM. Dado $\epsilon > 0$, se parte del siguiente problema:

$$\begin{aligned} \operatorname{argmín} \quad & \sum_{i=1}^n \ln(|x_i| + \epsilon). & (\text{log-sum}) \\ x \in \mathbb{R}^n \quad & \\ \Phi x = b \quad & \end{aligned}$$

Este puede ser pensado como una buena alternativa a (P_0) , incluso mejor que la basada en la norma ℓ_1 . En efecto, como se muestra en la Figura 4.3, la función $\frac{1}{C} \ln\left(\frac{|x|}{\epsilon} + 1\right)$, con constante $C = \ln\left(\frac{1}{\epsilon} + 1\right)$, es una mejor aproximación que ℓ_1 de la pseudo-norma ℓ_0 . La desventaja es que no es una función convexa.

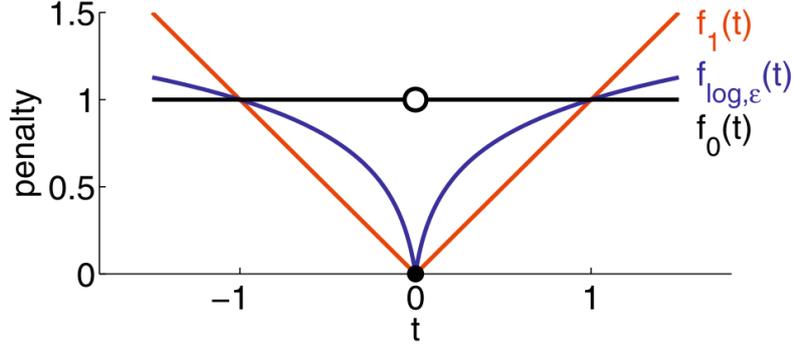


Figura 4.3: Funciones ℓ_0 (negro), ℓ_1 (rojo) y $\frac{1}{C} \ln \left(\frac{|x|}{\epsilon} + 1 \right)$ (azul). Esta última es una mejor aproximación de ℓ_0 que la dada por ℓ_1 . Sin embargo no es convexa. (Imagen tomada de [20]).

Introduciendo una variable auxiliar, el problema anterior se puede plantear en forma equivalente como:

$$\begin{aligned} \text{argmín} \quad & \sum_{i=1}^n \ln(u_i + \epsilon). \\ x, u \in \mathbb{R}^n \quad & \\ \Phi x = b \quad & \\ |x_i| \leq u_i, \forall i \quad & \end{aligned} \quad (4.5)$$

La función objetivo $g(x, u) = \sum_{i=1}^n \ln(u_i + \epsilon)$ resulta cóncava. Se puede aplicar entonces el algoritmo MM para estimar una solución de este último problema, hallando el mayorizante mediante linealización:

$$\begin{aligned} (x^{k+1}, u^{k+1}) \in \text{argmín} \quad & \sum_{i=1}^n \left(\frac{1}{u_i^k + \epsilon} \right) u_i. \\ x, u \in \mathbb{R}^n \quad & \\ \Phi x = b \quad & \\ |x_i| \leq u_i, \forall i \quad & \end{aligned} \quad (4.6)$$

Eliminando la variable auxiliar, este último problema resulta equivalente a:

$$\begin{aligned} x^{k+1} \in \text{argmín} \quad & \sum_{i=1}^n \left(\frac{1}{|x_i^k| + \epsilon} \right) |x_i|. \\ x \in \mathbb{R}^n \quad & \\ \Phi x = b \quad & \end{aligned} \quad (4.7)$$

Se recupera de esta forma el algoritmo $\text{RW}\ell_1$ de CWB, donde los pesos son actualizados mediante:

$$w_i^k = \frac{1}{|x_i^k| + \epsilon}.$$

Es decir que el algoritmo $\text{RW}\ell_1$ de CWB se puede obtener aplicando la metodología MM a una función cóncava. Es de esperarse entonces que la convergencia a un mínimo global de ([log-sum](#)) dependa de la condición inicial. En [20] se propone utilizar como condición inicial una solución del problema sin pesos (P_1). Esto equivale a iniciar el algoritmo $\text{RW}\ell_1$ con pesos unidad: $w^0 = \vec{1}$.

4.3. Desempeño bajo ruido acotado

Es posible establecer un resultado sobre el desempeño del algoritmo $\text{RW}\ell_1$ de CWB en presencia de ruido acotado, similar al obtenido en el Teorema 11 para minimización ℓ_1 . Por simplicidad se considera únicamente el caso en que la señal de interés es k -esparsa, aunque el teorema se puede extender al caso general [52][Teorema 3.2].

Teorema 13. ([52][Teorema 3.1]) *Supongamos que Φ satisface la propiedad RIP de orden $2k$, con constante $\delta_{2k} < \sqrt{2} - 1$. Se dispone de medidas: $b = \Phi x^* + z$, con $x^* \in \Sigma_k$ y ruido acotado: $\|z\|_2 \leq \eta$. Supongamos que la coordenada no nula de x^* con menor magnitud verifica:*

$$\min_{x_i^* \neq 0} |x_i^*| \geq \left(4 \frac{\sqrt{1 + \delta_{2k}}}{1 - (\sqrt{2} + 1)\delta_{2k}} \right) 2\eta = 2C_2\eta.$$

Sea $\hat{x} = \lim_{k \rightarrow \infty} x^k$, con x^k sucesión generada mediante $\text{RW}\ell_1$ de CWB. Entonces:

$$\|\hat{x} - x^*\|_2 \leq C_3\eta, \quad C_3 = 4 \frac{\sqrt{1 + \delta_{2k}}}{1 + (\sqrt{2} - 1)\delta_{2k}}.$$

En particular, si el ruido es nulo ($\eta = 0$), se logra recuperar x^* mediante $\text{RW}\ell_1$.

La condición sobre la coordenada no nula de x^* de menor magnitud introduce un mínimo necesario en la “relación señal a ruido” del problema. En principio este teorema no resulta revelador pues, bajo hipótesis similares, el Teorema 11 parece garantizar igual desempeño de la minimización ℓ_1 respecto al ruido. Sin embargo, contrario a lo que ocurría en los resultados de minimización ℓ_1 , ahora la constante C_3 se mantiene acotada para todo $\delta_{2k} \in [0, 1)$, y en particular cuando $\delta_{2k} \rightarrow \sqrt{2} - 1$:

$$\lim_{\delta_{2k} \rightarrow (\sqrt{2}-1)^-} C_3 = \frac{2^{\frac{5}{4}}}{2 - \sqrt{2}} \simeq 4.06.$$

Como contrapartida se agrega la hipótesis de que las magnitudes no nulas de x^* superen un cierto umbral; el cual tiende a infinito cuando $\delta_{2k} \rightarrow (\sqrt{2} - 1)^-$. De todas formas, en el caso en que x^* tenga coordenadas no nulas demasiado pequeñas como para cumplir esta hipótesis, es posible aplicar el teorema tratando a dichas coordenadas como ruido [52].

4.4. Otros algoritmos RW

4.4.1. $\text{RW}\ell_2$

En la literatura de CS existen otros algoritmos de tipo RW, que no se basan en la norma ℓ_1 . Por ejemplo en [25] se analiza el desempeño de un algoritmo $\text{RW}\ell_2$, denominado “Iteratively Re Weighted

Least Squares” (IRLS). Este consiste en minimizar en cada paso un problema del tipo Weighted- ℓ_2 :

$$\begin{aligned} x^{k+1} \in \operatorname{argmín} & \quad \frac{1}{2} \sum_{i=1}^n w_i^k x_i^2 = W_k^{-1} \Phi^T (\Phi W_k^{-1} \Phi^T)^{-1} b, \\ \Phi x & = b \\ x & \in \mathbb{R}^n \end{aligned} \quad (P_2W)$$

a la vez que se actualizan los pesos mediante:

$$w_i^{k+1} = \frac{1}{\sqrt{(x_i^{k+1})^2 + \epsilon_k^2}}, \quad \forall i = 1, 2, \dots, n.$$

Una ventaja de $\text{RW}\ell_2$ frente a $\text{RW}\ell_1$ es que la solución de (P_2W) se puede hallar de forma explícita; mientras que para resolver el correspondiente problema (P_1W) asociado a cada paso de $\text{RW}\ell_1$, es necesario utilizar un algoritmo de optimización. Esto permite realizar una mayor cantidad de iteraciones de $\text{RW}\ell_2$, en un mismo límite de tiempo. Sin embargo, dado que la forma de la bola generada por la norma ℓ_1 favorece la obtención de soluciones esparsas, es de esperarse que se requieran menos iteraciones de $\text{RW}\ell_1$ para hallar una solución esparsa, en relación a las requeridas con $\text{RW}\ell_2$. A su vez, si se desean agregar restricciones al problema CS, como ser que la solución buscada sea no negativa, muchas veces se pierde la ventaja de disponer de una expresión explícita para la solución de cada problema Weighted ℓ_2 , mientras que en el algoritmo $\text{RW}\ell_1$ la cantidad de operaciones por iteración no se ve demasiado afectada con la incorporación de restricciones.

4.4.2. $\text{RW}\ell_2$ regularizado

En [22], Chartrand y Yin proponen modificar el valor del parámetro ϵ_k de $\text{RW}\ell_2$ solamente cuando la variación en x^k sea considerable. En concreto proponen el siguiente procedimiento:

- se inicia con $\epsilon_0 = 1$.
- si $\frac{\|x^{k+1} - x^k\|_2}{\|x^k\|_2} < \sqrt{\epsilon_i} \frac{1}{100} \Rightarrow \epsilon_{i+1} = \epsilon_i \frac{1}{10}$.

Notar que el índice k de la iteración $\text{RW}\ell_1$ no necesariamente coincide con el índice i de la regularización. Como forma de evitar problemas numéricos al calcular W_k^{-1} , la iteración de $\text{RW}\ell_2$ es finalizada si $\epsilon_i < 1 \times 10^{-8}$. Como se puede intuir, para que este proceso de regularización sea útil, es necesario realizar una cantidad importante de iteraciones de RW. Esto hace que el proceso sea más adecuado para algoritmos como $\text{RW}\ell_2$, donde el costo computacional de cada iteración de RW es bajo. Nos referiremos a esta forma de actualizar el parámetro ϵ como método RW regularizado.

Los experimentos numéricos realizados en [22], donde se considera un problema CS aleatorio de tamaño $n = 256$ y $m = 100$, con medidas sin ruido y valores no nulos $x_i^* \sim N(0, \sigma = 2)$, indican que esta regularización logra una mejora considerable en el desempeño del algoritmo $\text{RW}\ell_2$. En dichas pruebas se implementa además el algoritmo $\text{RW}\ell_1$ de CWB con igual proceso de regularización de su parámetro ϵ , obteniéndose un desempeño muy similar entre ambos algoritmos regularizados. Los autores no especifican la cantidad de iteraciones realizadas en sus pruebas con cada algoritmo.

Capítulo 5

Algoritmo $RW\ell_1$ propuesto (sin ruido)

En este capítulo se propone una nueva metodología para actualizar los pesos del algoritmo $RW\ell_1$. Esta se basa en identificar dichos pesos con los multiplicadores de Lagrange de un problema convexo. A partir de dicha metodología, y utilizando el método subgradiente proyectado para estimar soluciones del problema dual, se obtiene una nueva versión del algoritmo $RW\ell_1$. A lo largo del capítulo se utilizan algunos conceptos y resultados vinculados a la teoría de la relajación lagrangeana. Se pueden encontrar varias referencias de introducción al tema, como por ejemplo [12] y [7].

5.1. Problema primal con oráculo

Para introducir la metodología propuesta, se considera el caso ideal en que se conoce una solución x^* de (P_0) . A partir de esta solución se puede definir el siguiente problema (P) , al cual denominaremos problema primal con oráculo y cuyo valor óptimo es $f^* = 0$:

$$\begin{aligned} & \text{argmín} && 0. && (P) \\ & \Phi x = b \\ & |x_i| \leq |x_i^*|, \forall i = 1, 2, \dots, n \\ & x \in \mathbb{R}^n \end{aligned}$$

Dado que la función objetivo es constante, (P) se puede ver también como un problema de factibilidad, en el cual se busca un conjunto de puntos que cumplan ciertas desigualdades. Claramente x^* es solución de (P) pues cumple todas las restricciones de factibilidad. Por otro lado, se ve fácilmente que si \hat{x} es solución de (P) , su soporte coincide con el de x^* , por lo que debe ser solución de (P_0) . Por lo tanto: si (P_0) tiene solución única, los problemas (P) y (P_0) son equivalentes.

5.2. Problema relajado

Veamos ahora que si se considera una relajación Lagrangeana de (P) , se obtiene un problema de tipo (P_1W) . Relajando las restricciones ideales de (P) , dadas por $|x_i| \leq |x_i^*|$, se obtiene el siguiente

lagrangeano:

$$L(x, w) = \sum_{i=1}^n w_i (|x_i| - |x_i^*|) = \sum_{i=1}^n w_i |x_i| - \sum_{i=1}^n w_i |x_i^*|. \quad (5.1)$$

La función dual asociada es entonces:

$$d(w) = \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} L(x, w) = \left(\min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \sum_{i=1}^n w_i |x_i| \right) - \sum_{i=1}^n w_i |x_i^*|. \quad (5.2)$$

Este es un problema de tipo “Weighted ℓ_1 ”, donde los pesos $w_i \geq 0$ son los multiplicadores de la relajación de Lagrange. Esto permite colocar el problema (P_1W) en el marco de la teoría bien conocida de la relajación de Lagrange. En particular permite estimar los pesos mediante algoritmos destinados a estimar multiplicadores de Lagrange.

5.3. Problema dual

En muchos casos los multiplicadores de Lagrange coinciden con las soluciones del problema dual asociado, por lo que también es posible estimar los pesos mediante algoritmos destinados a estimar soluciones del dual. Por definición el problema dual está dado por:

$$\begin{aligned} \operatorname{argm\acute{a}x} \quad & d(w). \\ & w \geq 0 \\ & w \in \mathbb{R}^n \end{aligned} \quad (D)$$

5.4. Soluciones del dual

Antes de presentar algoritmos para estimar soluciones del dual, conviene analizar cómo son estas soluciones. Veamos primero que (D) siempre admite al menos una solución. Por dualidad débil se tiene:

$$d(w) \leq f^* = 0, \quad \forall w \geq 0.$$

Por otro lado, si se consideran los siguientes pesos ideales finitos: $w_i^* = \begin{cases} 1, & x_i^* = 0 \\ 0, & x_i^* \neq 0 \end{cases}$, se tiene:

$$d(w^*) = \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} L(x, w^*) = \left(\min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \sum_{i / x_i^*=0} |x_i| \right) - \sum_{i / x_i^*=0} |x_i^*| = 0 - 0 = 0 = f^*.$$

Es decir: existe $w^* \geq 0$, y por lo tanto dual factible, en el cual la función dual alcanza la cota

superior de dualidad débil. Por lo tanto: w^* es solución del problema dual, el valor óptimo dual es $d^* = 0$ y no hay gap de dualidad: $d^* = f^*$. Más en general, cualquier vector de pesos $\hat{w} \geq 0$, cuyas coordenadas sean nulas en el soporte de x^* , será solución del problema dual:

Proposición 1. *Sea $\hat{w} \geq \vec{0}$ y tal que $\hat{w}_i = 0, \forall x_i^* \neq 0$. Entonces \hat{w} es solución del problema dual (D).*

Demostración. Como por hipótesis \hat{w} es dual factible, sólo resta probar que $d(\hat{w}) = d^* = 0$. Por definición:

$$d(\hat{w}) = \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} L(x, \hat{w}) = \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \left(\min_{i / x_i^* = 0} \sum \hat{w}_i |x_i| \right) - \sum_{i / x_i^* = 0} \hat{w}_i |x_i^*| = 0 - 0 = 0.$$

□

Como lo establece el siguiente resultado, en el caso más restrictivo en que el soporte de \hat{w} coincide con el complemento del soporte de x^* , se puede obtener una solución de CS a partir del problema relajado.

Proposición 2. *Sea $\hat{w} \geq 0$ que verifica: $\hat{w}_i = 0 \Leftrightarrow x_i^* \neq 0$. Consideremos ahora:*

$$\hat{x} \in \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\text{argmín}} L(x, \hat{w}) = \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\text{argmín}} \sum_{i=1}^n \hat{w}_i |x_i|.$$

Entonces \hat{x} es solución del problema (P_0) de CS.

Demostración. Por definición de \hat{x} y usando que $\hat{w}_i = 0, \forall x_i^* \neq 0$, se tiene:

$$\hat{x} \in \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\text{argmín}} \sum_{i=1}^n \hat{w}_i |x_i| = \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\text{argmín}} \sum_{i / x_i^* = 0} \hat{w}_i |x_i|.$$

Como $\hat{w}_i \geq 0$, el menor valor que puede tomar la función objetivo es el nulo, el cual se alcanza en x^* :

$$\sum_{i / x_i^* = 0} \hat{w}_i |x_i| \geq 0 = \sum_{i / x_i^* = 0} \hat{w}_i |x_i^*|.$$

Por lo tanto, el punto \hat{x} donde se alcanza el óptimo también debe anular la suma:

$$\sum_{i / x_i^* = 0} \hat{w}_i |\hat{x}_i| = 0.$$

Como $\hat{w}_i > 0$ para todo $x_i^* = 0$, se concluye que: $\hat{x}_i = 0$ para todo i tal que $x_i^* = 0$. En particular: $\|\hat{x}\|_0 \leq \|x^*\|_0$. La prueba concluye notando que \hat{x} es factible en (P_0) por definición: $\Phi\hat{x} = b$. \square

Este resultado es muy alentador pues, si todas las soluciones w del dual tuvieran como soporte al complemento del soporte de x^* , bastaría con encontrar una solución cualquiera del dual para resolver CS de forma eficiente. Sin embargo, aún en este caso ideal en que se conoce x^* , existen soluciones del dual cuyo soporte se intersecta con el de x^* . En efecto, es fácil ver que el vector nulo es solución del dual, pues este es dual factible y cumple:

$$d(\vec{0}) = \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \sum_{i=1}^n 0 (|x_i| - |x_i^*|) = 0 = d^*.$$

5.5. Algoritmo con oráculo (usando método subgradiente proyectado)

En las secciones anteriores se introdujo la metodología propuesta para estimar los pesos. Para obtener un algoritmo a partir de dicha metodología, es necesario seleccionar algún algoritmo que permita estimar multiplicadores de Lagrange o soluciones del dual. Para esto último se utilizará el “método subgradiente proyectado”.

5.5.1. Método subgradiente proyectado

El problema dual (D) consiste en maximizar una función que es siempre cóncava:

$$\begin{aligned} & \operatorname{argmáx} \quad d(w). \\ & w \geq 0 \\ & w \in \mathbb{R}^n \end{aligned} \tag{D}$$

Un posible algoritmo para estimar una solución de (D) es el “método subgradiente proyectado”. Este se puede pensar como un “ascenso” por (sub)gradiente en la función dual, la cual es cóncava, seguido de una proyección en la región factible $w \geq 0$ [7][Sección 6.3.1]:

$$\begin{cases} w^0 \geq 0 \\ w^{k+1} = w^k + \alpha_k g^k, \alpha_k > 0, g^k \in \partial d(w^k) \\ w^{k+1} = \operatorname{máx}\{0, w^{k+1}\} \end{cases} .$$

Cabe aclarar, sin embargo, que el método subgradiente proyectado no es un método de ascenso. Es decir que en alguna iteración k puede ocurrir:

$$d(w^{k+1}) < d(w^k), \forall \alpha_k > 0.$$

Esto puede darse en los puntos donde la función dual es no diferenciable, tal como se muestra en la Figura 5.1.

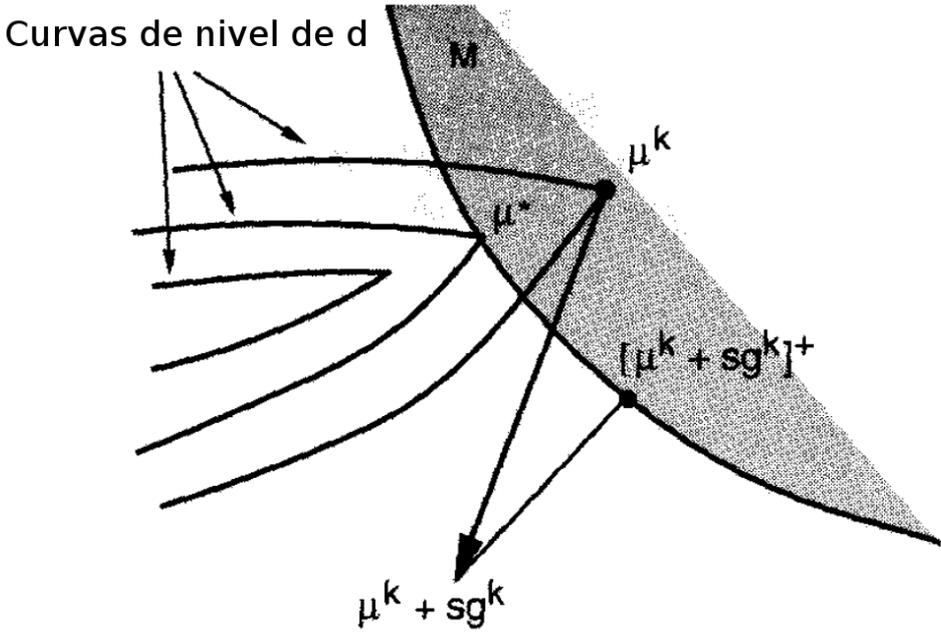


Figura 5.1: Curvas de nivel de una función dual no diferenciable d . El conjunto dual factible se representa mediante la región sombreada. En el punto μ^k , donde d es no diferenciable, no es posible moverse en la dirección del subgradiente g^k seleccionado sin disminuir el valor funcional. (Imagen de [7][Sección 6.3.1]).

Sin embargo, mientras no se alcance el óptimo, siempre es posible elegir el paso de forma que la distancia al conjunto solución S_d decretezca [7][Proposición 6.3.1]:

$$\forall k \geq 0, w^* \in S_d, \exists \alpha_k > 0 / \|w^{k+1} - w^*\| < \|w^k - w^*\|.$$

5.5.2. Cálculo del subgradiente

En cada paso del método subgradiente proyectado se requiere un subgradiente g^k de la función dual. Para esto se puede utilizar el siguiente resultado:

Teorema 14. ([7][Sección 6.1]) Consideremos el siguiente problema primal convexo, con función objetivo no necesariamente diferenciable:

$$\begin{aligned} & \text{mín} && f(x). \\ & g_i(x) \leq 0, \forall i = 1, \dots, m \\ & x \in X \end{aligned}$$

Sea \hat{x} tal que:

$$\hat{x} \in \underset{x \in X}{\text{argmín}} L(x, \hat{w}) = \underset{x \in X}{\text{argmín}} \{f(x) + \hat{w}^T g(x)\}.$$

Entonces $g(\hat{x}) \in \partial d(\hat{w})$; donde $\partial d(\hat{w})$ es el conjunto conformado por los subgradientes de la

función dual en \hat{w} , denominado subdiferencial. Esta última función viene dada por:

$$d(w) := \min_{x \in X} \{f(x) + w^T g(x)\}.$$

En nuestro caso de interés el algoritmo subgradiente proyectado se aplica a la función dual asociada al problema primal (P) , para el cual:

$$f(x) = 0, \forall x, \quad g_i(x) = |x_i| - |x_i^*|, \quad X = \{x \in \mathbb{R}^n / \Phi x = b\}.$$

Por lo tanto, para obtener un subgradiente mediante el teorema anterior, se debe resolver un problema Weighted- ℓ_1 .

5.5.3. Selección del paso

Una forma de seleccionar el paso, que garantiza convergencia del algoritmo subgradiente a una solución del dual, es [43][Teorema 5.1.2]:

$$\alpha_k = \frac{\beta_k}{\|g^k\|_2}, \quad \beta_k = \frac{1}{k+1}, \quad g^k \in \partial d(w^k).$$

De esta forma se obtiene convergencia de orden sub-lineal en las variables. Otra opción, que en algunos casos permite mejorar el orden de convergencia, consiste en tomar [43][Sección 5.1]:

$$\alpha_k = \frac{d^* - d(w^k)}{\|g^k\|_2^2} = \frac{d^* - L(x^{k+1}, w^k)}{\|g(x^{k+1})\|_2^2} \geq 0.$$

Esta última opción requiere conocer el valor óptimo d^* del dual. En el caso ideal bajo consideración, este es conocido y vale $d^* = f^* = 0$. Por lo tanto resulta:

$$\alpha_k = \frac{0 - L(x^{k+1}, w^k)}{\|g(x^{k+1})\|_2^2} = -\frac{\sum_{i=1}^n w_i^k (|x_i^{k+1}| - |x_i^*|)}{\sum_{i=1}^n (|x_i^{k+1}| - |x_i^*|)^2} \geq 0, \quad \forall k \geq 0. \quad (5.3)$$

5.5.4. Algoritmo propuesto

En el Algoritmo 2 se muestra un pseudocódigo del algoritmo obtenido al combinar el método subgradiente proyectado con la metodología propuesta. Se obtiene así un algoritmo $RW\ell_1$ que, en cada paso resuelve un problema (P_1W) , y luego actualiza los pesos w^k moviéndose en la dirección y sentido de un subgradiente. Al igual que para el $RW\ell_1$ de CWB, el algoritmo propuesto se implementa de forma que las iteraciones realizadas sean literalmente iteraciones de Re-Weighted.

Algorithm 2 $\text{RW}\ell_1$ con pesos actualizados mediante subgradiente en dual (con oráculo y sin ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\text{maxIter} \geq 1$

- 1: $x^1 \in \text{argmín}_{x \in \mathbb{R}^n} \Phi x = b \quad \sum_{i=1}^n w_i^0 |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^0\}$
 - 2:
 - 3: $k = 0$
 - 4: **while** $k < \text{maxIter}$ **do**
 - 5: $g_i^k = g_i(x^{k+1}) = |x_i^{k+1}| - |x_i^*|$ {cálculo de un subgradiente}
 - 6:
 - 7: calcular α_k usando (5.3)
 - 8:
 - 9: $w_i^{k+1} = w_i^k + \alpha_k g_i^k$
 - 10: $w_i^{k+1} = \text{máx}(0, w_i^{k+1})$ {proyección sobre $w \geq 0$ }
 - 11:
 - 12: $k = k + 1$
 - 13:
 - 14: $x^0 = x^k$
 - 15: $x^{k+1} \in \text{argmín}_{x \in \mathbb{R}^n} \Phi x = b \quad \sum_{i=1}^n w_i^k |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^k\}$
 - 16:
 - 17: **end while**
 - 18: **return** x^{k+1}
-

En el paso k la coordenada i -ésima de los pesos se actualiza mediante la proyección en \mathbb{R}^+ de:

$$w_i^{k+1} = w_i^k + \alpha_k g_i(x^{k+1}) = w_i^k + \alpha_k (|x_i^{k+1}| - |x_i^*|) = \begin{cases} > w_i^k & \text{si } |x_i^{k+1}| > |x_i^*| \\ < w_i^k & \text{si } |x_i^{k+1}| < |x_i^*| \\ = w_i^k & \text{si } |x_i^{k+1}| = |x_i^*| \end{cases} .$$

Por lo tanto: si la coordenada i -ésima x_i^{k+1} no cumple la restricción primal ideal, dada por $|x_i^{k+1}| \leq |x_i^*|$, su peso aumenta, fomentando que en el próximo paso su magnitud decrezca. De forma similar, si x_i^{k+1} cumple la restricción primal de forma estricta: $|x_i^{k+1}| < |x_i^*|$, su peso disminuye, permitiendo que en el próximo paso aumente su magnitud.

5.6. Algoritmo sin oráculo

En el caso en que no se conoce una solución x^* de (P_0) , es necesario reemplazar las restricciones ideales de (P) , dadas por $g_i(x) = |x_i| - |x_i^*|$, por una aproximación de las mismas. Una forma sencilla de hacerlo es reemplazar x^* por su estimación actual x^{k+1} (“amplificada”), obteniendo así para cada k una función g^k con coordenadas:

$$g_i^k(x) = |x_i| - (1 + \epsilon_k) |x_i^{k+1}|, \quad \epsilon_k > 0.$$

Donde, al igual que antes:

$$x^{k+1} \in \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\operatorname{argmín}} \sum_{i=1}^n w_i^k |x_i|.$$

5.7. Problema primal sin oráculo

Teniendo en cuenta la forma de las restricciones no ideales propuestas, a cada paso del algoritmo se le puede asociar un problema primal sin oráculo, dado por:

$$\begin{aligned} & \underset{\substack{\Phi x = b \\ |x_i| \leq (1 + \epsilon_k) |x_i^{k+1}|, \forall i \\ x \in \mathbb{R}^n}}{\operatorname{argmín}} & 0. \end{aligned} \quad (P^k)$$

Notar que por definición x^{k+1} verifica $\Phi x = b$. Por otro lado, se obtiene trivialmente que también verifica la restricción no ideal. Es decir que x^{k+1} siempre es solución del problema (P^k) y el valor óptimo de dicho problema es claramente $f^k = 0$.

5.8. Problema relajado sin oráculo

Relajando las restricciones no ideales de (P^k) se obtiene el siguiente lagrangeano:

$$L^k(x, w) = \sum_{i=1}^n w_i (|x_i| - (1 + \epsilon_k) |x_i^{k+1}|) = \sum_{i=1}^n w_i |x_i| - \sum_{i=1}^n w_i (1 + \epsilon_k) |x_i^{k+1}|. \quad (5.4)$$

Su función dual d^k es entonces un problema de tipo Weighted ℓ_1 , al igual que en el caso con oráculo:

$$d^k(w) = \underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\operatorname{mín}} L^k(x, w) = \left(\underset{\substack{\Phi x = b \\ x \in \mathbb{R}^n}}{\operatorname{mín}} \sum_{i=1}^n w_i |x_i| \right) - \sum_{i=1}^n w_i (1 + \epsilon_k) |x_i^{k+1}|. \quad (5.5)$$

5.9. Problema dual sin oráculo

El problema dual asociado es por definición:

$$\begin{aligned} & \operatorname{argmáx} & d^k(w). \\ & w \geq 0 \\ & w \in \mathbb{R}^n \end{aligned} \quad (D^k)$$

5.10. Soluciones del dual

Se analizarán ahora las posibles soluciones de cada problema dual sin oráculo (D^k).

Proposición 3. *El problema (D^k) siempre tiene solución. Además hay dualidad fuerte: $d^k = f^k = 0$.*

Demostración. Al igual que en el caso con oráculo, se define el siguiente candidato a solución del dual (D^k), el cual es en particular dual factible:

$$\hat{w}_i := \begin{cases} 1, & \text{si } x_i^{k+1} = 0 \\ 0, & \text{si } x_i^{k+1} \neq 0 \end{cases}.$$

Por dualidad débil se tiene la cota: $d^k(w) \leq f^k = 0, \forall w \geq 0$. Basta entonces con probar que en \hat{w} se alcanza dicha cota. Se tiene:

$$\begin{aligned} d^k(\hat{w}) &= \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} L^k(x, \hat{w}) = \left(\min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \sum_{i / x_i^{k+1} = 0} |x_i| \right) - \sum_{i / x_i^{k+1} = 0} (1 + \epsilon_k) |x_i^{k+1}| = \\ &= \min_{\substack{\Phi x = b \\ x \in \mathbb{R}^n}} \sum_{i / x_i^{k+1} = 0} |x_i| = 0 = f^k. \end{aligned}$$

Por lo tanto \hat{w} es solución de (D^k) y además hay dualidad fuerte: $d^k = f^k = 0$. □

Observación 7. *Al igual que en el caso con oráculo, el vector nulo es solución de cada dual (D^k), pues: $d^k(\vec{0}) = 0 = d^k$.*

Dado que el valor dual óptimo de cada paso es $d^k = 0$, es posible utilizar el paso óptimo en la implementación del algoritmo subgradiente proyectado, aplicado a cada problema dual (D^k). Definiendo la matriz diagonal $W^k = \text{diag}(w^k)$, este paso se puede expresar como:

$$\alpha_k = \frac{d^k - d^k(w^k)}{\|g^k\|_2^2} = \frac{0 - L^k(x^{k+1}, w^k)}{\|g^k(x^{k+1})\|_2^2} = \frac{1 \|W^k x^{k+1}\|_1}{\epsilon_k \|x^{k+1}\|_2^2} \geq 0, \forall k \geq 0. \quad (5.6)$$

En el Algoritmo 3 se muestra un pseudocódigo del algoritmo $\text{RW}\ell_1$ sin oráculo, obtenido al combinar la metodología propuesta con el método subgradiente proyectado.

Algorithm 3 $RW\ell_1$ con pesos actualizados mediante subgradiente en dual (sin oráculo y sin ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\text{maxIter} \geq 1$

```
1:
2:  $x^1 \in \operatorname{argmín}_{x \in \mathbb{R}^n} \Phi x = b \quad \sum_{i=1}^n w_i^0 |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^0\}$ 
3:
4:  $k = 0$ 
5: while  $k < \text{maxIter}$  do
6:
7:    $g_i^k = g_i^k(x^{k+1}) = |x_i^{k+1}| - (1 + \epsilon_k) |x_i^{k+1}| = -\epsilon_k |x_i^{k+1}|$  {cálculo de subgradiente no ideal}
8:
9:   calcular  $\alpha_k \geq 0$  usando (5.6)
10:
11:    $w_i^{k+1} = \operatorname{máx}(0, w_i^k + \alpha_k g_i^k)$ 
12:
13:    $k = k + 1$ 
14:
15:    $x^0 = x^k$ 
16:    $x^{k+1} \in \operatorname{argmín}_{x \in \mathbb{R}^n} \Phi x = b \quad \sum_{i=1}^n w_i^k |x_i| \quad \{(P_1 W) \text{ con condición inicial } x^k\}$ 
17:
18: end while
19: return  $x^{k+1}$ 
```

En el paso k la coordenada i -ésima de los pesos se actualiza mediante la proyección en \mathbb{R}^+ de:

$$w_i^{k+1} = w_i^k + \alpha_k g_i^k(x^{k+1}) = w_i^k + \left(\frac{1}{\epsilon_k} \frac{\|W^k x^{k+1}\|_1}{\|x^{k+1}\|_2^2} \right) (-\epsilon_k |x_i^{k+1}|) = w_i^k - \frac{\|W^k x^{k+1}\|_1}{\|x^{k+1}\|_2^2} |x_i^{k+1}|.$$

Contrario a lo que ocurría en el caso con oráculo, en este caso los pesos w_i sólo pueden decrementar. Se observa a su vez que el valor del parámetro $\epsilon_k > 0$ es irrelevante en la actualización de los w_i . Más aún, el valor de ϵ_k es irrelevante en todo el Algoritmo (3), siempre que sea estrictamente positivo.

Capítulo 6

Algoritmo $\text{RW}\ell_1$ propuesto (con ruido)

En este capítulo se extiende la metodología propuesta en el Capítulo 5, y su correspondiente algoritmo $\text{RW}\ell_1$ con subgradiente proyectado, al caso en que las medidas b están afectadas por ruido Gaussiano:

$$b = \Phi x + z, \text{ con coordenadas independientes } z_i \sim N(0, \sigma^2).$$

En este caso se busca resolver el problema (P_0^η) , para cierto nivel de ruido $\eta \geq 0$:

$$\begin{aligned} \text{argmín} \quad & \|x\|_0. \\ & \|\Phi x - b\|_2 \leq \eta \\ & x \in \mathbb{R}^n \end{aligned} \tag{P_0^\eta}$$

El parámetro η se debe seleccionar de forma que la solución buscada x^* sea factible: $\|\Phi x^* - b\|_2 \leq \eta$ (con alta probabilidad). El problema Weighted ℓ_1 asociado es:

$$\begin{aligned} \text{argmín} \quad & \sum_{i=1}^n w_i |x_i|. \\ & \|\Phi x - b\|_2 \leq \eta \\ & x \in \mathbb{R}^n \end{aligned} \tag{P_1^\eta W}$$

Al igual que antes, el algoritmo $\text{RW}\ell_1$ busca estimar pesos $w_i \geq 0$, de forma que al resolver el problema asociado $(P_1^\eta W)$ se obtenga una solución de (P_0^η) .

6.1. Problema con oráculo

Para extender la metodología propuesta al problema con ruido, se inicia considerando nuevamente un problema primal con oráculo, en donde se asume que se conoce una solución x^* de (P_0) :

$$\begin{aligned} \text{argmín} \quad & 0. \\ & \|\Phi x - b\|_2 \leq \eta \\ & |x_i| \leq |x_i^*|, \forall i \\ & x \in \mathbb{R}^n \end{aligned} \tag{P^\eta}$$

Si se relajan las restricciones ideales, se obtiene el mismo lagrangeano que para el caso sin ruido:

$$L(x, w) = \sum_{i=1}^n w_i (|x_i| - |x_i^*|) = \sum_{i=1}^n w_i |x_i| - \sum_{i=1}^n w_i |x_i^*|. \quad (6.1)$$

La función dual asociada es ahora:

$$d(w) = \min_{\substack{\|\Phi x - b\|_2 \leq \eta \\ x \in \mathbb{R}^n}} L(x, w) = \left(\min_{\substack{\|\Phi x - b\|_2 \leq \eta \\ x \in \mathbb{R}^n}} \sum_{i=1}^n w_i |x_i| \right) - \sum_{i=1}^n w_i |x_i^*|. \quad (6.2)$$

Se obtiene entonces un problema de tipo Weighted ℓ_1 con ruido. Nuevamente, esto permite identificar los pesos w_i de $(P_1^\eta W)$ con los multiplicadores de Lagrange de un problema convexo.

6.2. Problema sin oráculo

Se considera ahora el caso no ideal en que no se conoce x^* . Las restricciones ideales de $(P_1^\eta W)$ coinciden con las del caso sin ruido $(P_1 W)$: $g_i(x) = |x_i| - |x_i^*|$. Esto permite utilizar el mismo tipo de aproximación que en dicho caso para reemplazarlas:

$$g_i^k(x) = |x_i| - (1 + \epsilon_k) |x_i^{k+1}|, \quad \epsilon_k > 0.$$

La única diferencia es que ahora se utiliza:

$$x^{k+1} \in \operatorname{argmín}_{\substack{\|\Phi x - b\|_2 \leq \eta \\ x \in \mathbb{R}^n}} \sum_{i=1}^n w_i^k |x_i|.$$

6.3. Algoritmo propuesto (sin oráculo y con método subgradiente proyectado)

En el Algoritmo (4) se muestra un pseudocódigo del algoritmo $RW\ell_1$ obtenido con la metodología propuesta, para el caso con ruido y sin oráculo. Se utiliza nuevamente el método subgradiente proyectado para estimar soluciones del problema dual. El algoritmo es idéntico al del caso sin ruido, salvo que ahora x^{k+1} es solución de un problema Weighted- ℓ_1 con ruido. En particular la forma del paso óptimo coincide con la del caso sin ruido:

$$\alpha_k = \frac{d^k - d^k(w^k)}{\|g^k\|_2^2} = \frac{0 - L^k(x^{k+1}, w^k)}{\|g^k(x^{k+1})\|_2^2} = \frac{1}{\epsilon_k} \frac{\|W^k x^{k+1}\|_1}{\|x^{k+1}\|_2^2} \geq 0, \quad \forall k \geq 0. \quad (6.3)$$

Valen entonces las mismas consideraciones sobre la actualización de los pesos que se obtuvieron

en el caso sin oráculo y sin ruido. Esto es: en el paso k los pesos w_i sólo pueden decrementar y el valor de $\epsilon_k > 0$ es irrelevante en todo el Algoritmo (4).

Algorithm 4 $\text{RW}\ell_1$ con pesos actualizados mediante subgradiente en dual (sin oráculo y con ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\text{maxIter} \geq 1$, $\eta \geq 0$

1:
2: $x^1 \in \underset{x \in \mathbb{R}^n}{\text{argmín}} \|\Phi x - b\|_2 \leq \eta \quad \sum_{i=1}^n w_i^0 |x_i| \quad \{(P_1^\eta W) \text{ con condición inicial } x^0\}$
3:
4: $k = 0$
5: **while** $k < \text{maxIter}$ **do**
6:
7: $g_i^k = g_i^k(x^{k+1}) = |x_i^{k+1}| - (1 + \epsilon_k) |x_i^{k+1}| = -\epsilon_k |x_i^{k+1}|$ {cálculo de subgradiente no ideal}
8:
9: elegir $\alpha_k \geq 0$ usando (6.3)
10:
11: $w_i^{k+1} = \text{máx}(0, w_i^k + \alpha_k g_i^k)$
12:
13: $k = k + 1$
14:
15: $x^0 = x^k$
16: $x^{k+1} \in \underset{x \in \mathbb{R}^n}{\text{argmín}} \|\Phi x - b\|_2 \leq \eta \quad \sum_{i=1}^n w_i^k |x_i| \quad \{(P_1^\eta W) \text{ con condición inicial } x^k\}$
17:
18: **end while**
19: **return** x^{k+1}

Capítulo 7

Implementación: algoritmos para resolver Weighted- ℓ_1

Para implementar los algoritmos $\text{RW}\ell_1$, ya sea el de CWB o el obtenido con la metodología propuesta, es necesario resolver en cada paso un problema Weighted- ℓ_1 . En este capítulo se analizan algunas posibles formas de hacerlo.

7.1. Problema sin ruido: Programación Lineal (LP)

Se considera inicialmente el problema Weighted- ℓ_1 sin ruido. Si bien este es convexo y con restricciones lineales, presenta la dificultad de tener función objetivo no diferenciable:

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{argmín}} \sum_{i=1}^n w_i |x_i|. & (P_1W) \\ & \Phi x = b \end{aligned}$$

Esto último impide el uso de algoritmos de optimización estándar, basados en el gradiente. Sin embargo, es posible reformularlo como un problema de programación lineal (LP), de la siguiente forma [16][Apéndice A]:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{argmín}} \sum_{i=1}^n w_i |x_i| &= \underset{x, u \in \mathbb{R}^n}{\text{argmín}} \sum_{i=1}^n w_i u_i &= \underset{x, u \in \mathbb{R}^n}{\text{argmín}} \sum_{i=1}^n w_i u_i. & (LP) \\ \Phi x = b & \Phi x = b & \Phi x = b \\ & |x_i| \leq |u_i| & x_i - u_i \leq 0 \\ & & -x_i - u_i \leq 0 \end{aligned}$$

Esto permite estimar soluciones mediante algoritmos eficientes, provenientes de las distintas variantes del “método de punto interior” [55], [12][Capítulo 11]. El costo a pagar es la introducción de variables auxiliares, duplicando así la cantidad de incógnitas a $2n$ y aumentando las restricciones de m a $m + 2n$.

7.2. Problema con ruido: Programación Cónica de Segundo Orden (SOCP)

La situación es similar en el problema Weighted- ℓ_1 con ruido, salvo que las restricciones son cuadráticas:

$$\begin{aligned} \text{argmín} \quad & \sum_{i=1}^n w_i |x_i|. & (P_1^\eta W) \\ \|\Phi x - b\|_2 \leq \eta \\ x \in \mathbb{R}^n \end{aligned}$$

Se puede reformular el problema en forma análoga al caso sin ruido, obteniendo en este caso una función objetivo lineal con restricciones cuadráticas [16][Apéndice D]:

$$\begin{aligned} \text{argmín} \quad & \sum_{i=1}^n w_i |x_i| = & \text{argmín} \quad & \sum_{i=1}^n w_i u_i = & \text{argmín} \quad & \sum_{i=1}^n w_i u_i. & (\text{SOCP}) \\ \|\Phi x - b\|_2 \leq \eta \\ x \in \mathbb{R}^n & & x, u \in \mathbb{R}^n & & x, u \in \mathbb{R}^n & & \\ & & \|\Phi x - b\|_2 \leq \eta & & \|\Phi x - b\|_2 \leq \eta & & \\ & & |x_i| \leq |u_i| & & x_i - u_i \leq 0 & & \\ & & & & -x_i - u_i \leq 0 & & \end{aligned}$$

Esta reformulación es de tipo “Second Order Cone Programming” (SOCP), lo cual permite el uso de algoritmos eficientes para estimar sus soluciones [28]. El costo a pagar es la introducción de n nuevas variables y $2n$ restricciones auxiliares.

En este trabajo las soluciones de $(P_1 W)$ y $(P_1^\eta W)$ se obtienen a partir de las respectivas reformulaciones descritas. Para resolver estas reformulaciones se utiliza el lenguaje de modelado “Convex” de Julia [69, 8], con el solver “ECOS”, que implementa algoritmos de tipo “punto interior” [28]. Todos estos paquetes forman parte del proyecto “JuliaOpt”.

Paquete ℓ_1 -MAGIC

La metodología descrita, basada en reformular el problema Weighted- ℓ_1 , es frecuente en la literatura de CS. Por ejemplo, Candès y Romberg, dos de los autores de los artículos seminales de CS, han puesto a disposición el paquete de software ℓ_1 -MAGIC, que permite estimar soluciones de $(P_1 W)$ y $(P_1^\eta W)$ mediante las mencionadas reformulaciones [16]. Para resolver estas formulaciones utilizan algoritmos genéricos de tipo “punto interior”. Si bien el mencionado paquete es muy popular, su objetivo no es ofrecer un algoritmo “cutting-edge”, sino mostrar que el proceso de recuperación ℓ_1 es escalable y permite resolver problemas del orden de un millón de datos.

7.3. Algoritmos específicos

Existen algoritmos que buscan resolver $(P_1 W)$ y $(P_1^\eta W)$ sin reformularlos como un LP o SOCP. En su lugar trabajan directamente con la norma ℓ_1 y su no diferenciabilidad, lo cual puede significar

una ventaja en el desempeño. Por lo general estos algoritmos se basan en hallar soluciones de un problema LASSO (“Least Absolute Shrinkage and Selection Operator”), el cual se define como [73]:

$$\underset{x \in \mathbb{R}^n}{\operatorname{argmín}} \quad \frac{1}{2} \|\Phi x - b\|_2^2 + \mu \|x\|_1, \quad \mu > 0. \quad (\text{LASSO})$$

En lugar de analizar uno de estos algoritmos, en el Capítulo 8 se presenta una variante de la metodología propuesta, donde se reemplaza el problema (P_1^*W) por uno del tipo (LASSO). A su vez en el Capítulo 9 se presenta un algoritmo para resolver este último problema de forma eficiente, haciendo uso de la norma ℓ_1 . El problema (LASSO) es bien conocido y fue introducido por Tibshirani en 1996, en el área de la estadística [66]. En 1998, Chen, Donoho y Saunders introducen el mismo problema en el contexto de representación de señales, bajo el nombre de Basis Pursuit Denoising [23].

Capítulo 8

Algoritmo RW-LASSO propuesto

En este capítulo se obtiene una nueva versión del algoritmo $RW\ell_1$ con ruido, que fuera propuesto en los Capítulos 5 y 6. Esta se basa en la misma metodología propuesta en dichos capítulos, implementada nuevamente mediante el método de subgradiente proyectado, pero aplicada ahora a un problema dual de tipo Weighted LASSO (W-LASSO), el cual se introduce a continuación.

8.1. Problema Weighted LASSO (W-LASSO)

Se considera nuevamente el problema primal con oráculo y con ruido, donde parte de las restricciones han sido elevadas al cuadrado por mayor conveniencia:

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{argmín}} && 0. && (P^n) \\ & \frac{1}{2} \|\Phi x - b\|_2^2 \leq \frac{\eta^2}{2} \\ & |x_i| \leq |x_i^*|, \forall i \end{aligned}$$

Si se procede igual que antes, relajando las restricciones ideales, se obtiene un problema de tipo $(P_1^n W)$, que permite identificar los pesos con multiplicadores de Lagrange:

$$d(w) = \left(\underset{x \in \mathbb{R}^n}{\text{mín}} \left(\frac{1}{2} \|\Phi x - b\|_2^2 \leq \frac{\eta^2}{2} \quad \sum_{i=1}^n w_i |x_i| \right) - \sum_{i=1}^n w_i |x_i^*|. \right) \quad (8.1)$$

Si se relaja además la restricción vinculada al ruido, se obtiene la siguiente función dual de variables $(w, \lambda) \in \mathbb{R}_+^{n+1}$:

$$d(w, \lambda) = \left(\underset{x \in \mathbb{R}^n}{\text{mín}} \left(\frac{\lambda}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i |x_i| \right) - \left(\frac{\lambda}{2} \eta^2 + \sum_{i=1}^n w_i |x_i^*| \right) \right). \quad (8.2)$$

Por lo tanto, para cada par (w, λ) fijo, el conjunto solución del problema a partir del cual se obtiene el valor dual coincide con el conjunto:

$$\operatorname{argmín}_{x \in \mathbb{R}^n} \frac{\lambda}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i |x_i|. \quad (\text{W-LASSO})$$

Este último problema se conoce como Weighted LASSO (W-LASSO) y constituye una generalización del problema LASSO, que fuera introducido al final del Capítulo 7. El problema dual asociado a esta relajación de (P^η) es:

$$\operatorname{máx}_{w \geq 0, \lambda \geq 0} d(w, \lambda) = \operatorname{máx}_{w \geq 0, \lambda \geq 0} \left\{ \left(\operatorname{mín}_{x \in \mathbb{R}^n} \frac{\lambda}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i |x_i| \right) - \left(\frac{\lambda}{2} \eta^2 + \sum_{i=1}^n w_i |x_i^*| \right) \right\}. \quad (D_\lambda^\eta)$$

Se puede ver fácilmente que este problema dual siempre admite solución y se tiene además dualidad fuerte. En efecto: el valor primal óptimo de (P^η) es $f^* = 0$. Por dualidad débil se tiene: $d(w, \lambda) \leq f^* = 0, \forall w \geq 0, \lambda \geq 0$ y por otro lado: $d(\vec{0}, 0) = 0$. Es decir que $(\vec{0}, 0)$ es solución y el valor dual óptimo es: $d^* = f^* = 0$.

8.2. Algoritmo RW-LASSO con oráculo

A partir de esta nueva función dual, y utilizando la misma metodología introducida en los Capítulos 5 y 6, se puede obtener una nueva versión de dicha metodología, donde ahora el problema (W-LASSO) juega el papel de $(P_1^\eta W)$. La misma consiste en actualizar los pesos del algoritmo RW estimando soluciones (w^*, λ^*) del nuevo problema dual. Notar en particular que los multiplicadores $w_i \geq 0$ juegan el mismo papel que antes, por lo que se pueden interpretar como los pesos a estimar en $(P_1^\eta W)$. El lagrangeano asociado al problema primal con oráculo es ahora:

$$L(x, w, \lambda) = \frac{\lambda}{2} (\|\Phi x - b\|_2^2 - \eta^2) + \sum_{i=1}^n w_i (|x_i| - |x_i^*|).$$

Por lo tanto, si se utiliza el algoritmo subgradiente proyectado para estimar soluciones de $d(w, \lambda)$, el paso mediante el cual se actualizan los multiplicadores es:

$$\alpha_k = \frac{0 - L(x^{k+1}, w^k, \lambda^k)}{\|g(x^{k+1})\|_2^2} = - \frac{\frac{\lambda^k}{2} (\|\Phi x^{k+1} - b\|_2^2 - \eta^2) + \sum_{i=1}^n w_i^k (|x_i^{k+1}| - |x_i^*|)}{\sum_{i=1}^n (|x_i^{k+1}| - |x_i^*|)^2 + \left(\frac{1}{2}\right)^2 (\|\Phi x^{k+1} - b\|_2^2 - \eta^2)^2} \geq 0, \forall k \geq 0. \quad (8.3)$$

En el Algoritmo (5) se muestra el algoritmo RW-LASSO obtenido al utilizar el método de subgradiente proyectado para estimar soluciones del dual. En cada paso se deben actualizar los pesos $w_i \geq 0$ y el nuevo multiplicador $\lambda \geq 0$. En el paso k la coordenada i -ésima de los pesos se actualiza

mediante la proyección en \mathbb{R}^+ de:

$$w_i^{k+1} = w_i^k + \alpha_k g_i(x^{k+1}) = w_i^k + \alpha_k (|x_i^{k+1}| - |x_i^*|) = \begin{cases} > w_i^k & \text{si } |x_i^{k+1}| > |x_i^*| \\ < w_i^k & \text{si } |x_i^{k+1}| < |x_i^*| \\ = w_i^k & \text{si } |x_i^{k+1}| = |x_i^*| \end{cases} .$$

Valen entonces las mismas consideraciones que para el algoritmo $\text{RW}\ell_1$ con oráculo y subgradiente proyectado, presentado en la Sección 5.5. Por otro lado el multiplicador λ se actualiza mediante la proyección en \mathbb{R}^+ de:

$$\lambda^{k+1} = \lambda^k + \alpha_k g_\lambda(x^{k+1}) = \lambda^k + \frac{\alpha_k}{2} (\|\Phi x^{k+1} - b\|_2^2 - \eta^2) = \begin{cases} > \lambda^k & \text{si } \|\Phi x^{k+1} - b\|_2^2 > \eta^2 \\ < \lambda^k & \text{si } \|\Phi x^{k+1} - b\|_2^2 < \eta^2 \\ = \lambda^k & \text{si } \|\Phi x^{k+1} - b\|_2^2 = \eta^2 \end{cases} .$$

Por lo tanto: si x^{k+1} no cumple con la restricción primal $\|\Phi x^{k+1} - b\|_2^2 \leq \eta^2$, el peso de dicha restricción, dada por el multiplicador λ_k , aumenta, fomentando que $\|\Phi x - b\|_2^2$ disminuya en la próxima iteración.

Algorithm 5 RW-LASSO con algoritmo subgradiente (con oráculo y con ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\lambda^0 \in \mathbb{R}$, $\text{maxIter} \geq 1$, $\eta \geq 0$

- 1:
 - 2: $x^1 \in \text{argmín}_{x \in \mathbb{R}^n} \frac{\lambda^0}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i^0 |x_i|$ {(W-LASSO) con condición inicial x^0 }
 - 3:
 - 4: $k = 0$
 - 5: **while** $k < \text{maxIter}$ **do**
 - 6: $g_i^k = g_i(x^{k+1}) = |x_i^{k+1}| - |x_i^*|$ {cálculo subgradiente restricciones ideales}
 - 7:
 - 8: $g_\lambda^k = g_\lambda(x^{k+1}) = \frac{1}{2} (\|\Phi x^{k+1} - b\|_2^2 - \eta^2)$ {cálculo subgradiente restricciones $\|\Phi x - b\|_2 \leq \eta$ }
 - 9:
 - 10: elegir $\alpha_k \geq 0$ utilizando (8.3)
 - 11:
 - 12: $w_i^{k+1} = \text{máx}(0, w_i^k + \alpha_k g_i^k)$ {actualiza multiplicadores}
 - 13: $\lambda^{k+1} = \text{máx}(0, \lambda^k + \alpha_k g_\lambda^k)$
 - 14:
 - 15: $k = k + 1$
 - 16:
 - 17: $x^0 = x^k$
 - 18: $x^{k+1} \in \text{argmín}_{x \in \mathbb{R}^n} \frac{\lambda^k}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i^k |x_i|$ {(W-LASSO) con condición inicial x^k }
 - 19:
 - 20: **end while**
 - 21: **return** x^{k+1}
-

8.3. Algoritmo RW-LASSO sin oráculo

Para obtener la versión sin oráculo del algoritmo RW-LASSO, se reemplazan como antes las restricciones ideales $g_i(x) = |x_i| - |x_i^*|$ por:

$$g_i^k(x) = |x_i| - (1 + \epsilon_k) |x_i^{k+1}|, \quad \epsilon_k > 0.$$

Donde ahora x^{k+1} es solución de (W-LASSO), con pesos w^k :

$$x^{k+1} \in \underset{x \in \mathbb{R}^n}{\operatorname{argmín}} \frac{\lambda^k}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i^k |x_i|.$$

El problema primal sin oráculo asociado al paso k es entonces:

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\operatorname{argmín}} && 0. && (P_k^\eta) \\ & \frac{1}{2} \|\Phi x - b\|_2^2 \leq \frac{\eta^2}{2} \\ & |x_i| \leq (1 + \epsilon_k) |x_i^{k+1}|, \quad \forall i \end{aligned}$$

La función dual asociada al paso k es:

$$d^k(w, \lambda) = \left(\underset{x \in \mathbb{R}^n}{\operatorname{mín}} \frac{\lambda}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i |x_i| \right) - \left(\frac{\lambda}{2} \eta^2 + \sum_{i=1}^n w_i (1 + \epsilon_k) |x_i^{k+1}| \right). \quad (8.4)$$

Se puede ver fácilmente que en este caso sin oráculo también se tiene dualidad fuerte. En efecto: el valor primal óptimo de (P_k^η) es $f^k = 0$. Por dualidad débil se tiene: $d^k(w, \lambda) \leq f^k = 0, \forall w \geq 0, \lambda \geq 0$ y por otro lado: $d^k(\vec{0}, 0) = 0$. Es decir que el valor dual óptimo es: $d^k = f^k = 0$. Esto permite tomar el paso de subgradiente proyectado como:

$$\alpha_k = \frac{0 - L^k(x^{k+1}, w^k, \lambda^k)}{\|g^k(x^{k+1})\|_2^2} = - \frac{\frac{\lambda^k}{2} (\|\Phi x^{k+1} - b\|_2^2 - \eta^2) - \epsilon_k \sum_{i=1}^n w_i^k |x_i^{k+1}|}{\epsilon_k^2 \sum_{i=1}^n (|x_i^{k+1}|)^2 + \left(\frac{1}{2}\right)^2 (\|\Phi x^{k+1} - b\|_2^2 - \eta^2)^2} \geq 0, \quad \forall k \geq 0. \quad (8.5)$$

En el Algoritmo (6) se muestra el algoritmo RW-LASSO obtenido al utilizar el método de subgradiente proyectado para estimar soluciones del dual en el caso sin oráculo. El problema (W-LASSO) juega ahora el papel que antes jugaba $(P_1^\eta W)$. La mayor diferencia entre estos dos problemas es que ahora la condición $\|\Phi x - b\|_2 \leq \eta$ se encuentra en la función objetivo, mientras que antes era parte de las restricciones.

Algorithm 6 RW-LASSO con algoritmo subgradiente (sin oráculo y con ruido)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$, $w^0 \geq 0$, $\lambda^0 \in \mathbb{R}$, $\maxIter \geq 1$, $\eta \geq 0$

```
1:
2:  $x^1 \in \operatorname{argmín}_{x \in \mathbb{R}^n} \frac{\lambda^0}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i^0 |x_i|$  {(W-LASSO) con condición inicial  $x^0$ }
3:
4:  $k = 0$ 
5: while  $k < \maxIter$  do
6:    $\epsilon_k = \frac{1}{k+1}$ 
7:    $g_i^k = g_i^k(x^{k+1}) = |x_i^{k+1}| - (1 + \epsilon_k) |x_i^{k+1}| = -\epsilon_k |x_i^{k+1}|$  {subgradiente restricciones no ideales}
8:
9:    $g_\lambda^k = g_\lambda^k(x^{k+1}) = \frac{1}{2} (\|\Phi x^{k+1} - b\|_2^2 - \eta^2)$  {subgradiente restricciones  $\|\Phi x - b\|_2$ }
10:
11:   elegir  $\alpha_k \geq 0$  utilizando (8.5)
12:
13:    $w_i^{k+1} = \operatorname{máx}(0, w_i^k + \alpha_k g_i^k)$  {actualiza multiplicadores}
14:    $\lambda^{k+1} = \operatorname{máx}(0, \lambda^k + \alpha_k g_\lambda^k)$ 
15:
16:    $k = k + 1$ 
17:
18:    $x^0 = x^k$ 
19:    $x^{k+1} \in \operatorname{argmín}_{x \in \mathbb{R}^n} \frac{\lambda^k}{2} \|\Phi x - b\|_2^2 + \sum_{i=1}^n w_i^k |x_i|$  {(W-LASSO) con condición inicial  $x^k$ }
20:
21: end while
22: return  $x^{k+1}$ 
```

La actualización de los pesos en el paso k es la proyección sobre \mathbb{R}^+ de:

$$w_i^{k+1} = w_i^k + \alpha_k g_i^k(x^{k+1}) = w_i^k - \alpha_k (\epsilon_k |x_i^{k+1}|).$$

Al igual que en el caso sin oráculo del algoritmo $\text{RW}\ell_1$ con subgradiente proyectado, los pesos w_i sólo pueden decrementar. La diferencia es que ahora el valor del parámetro $\epsilon_k > 0$ sí es relevante en la actualización de los w_i . La actualización del multiplicador λ^k es cualitativamente igual que en el caso con oráculo, cambiando únicamente el valor del paso α_k .

Capítulo 9

Implementación: algoritmos para resolver W-LASSO

En el Algoritmo (6) el problema (W-LASSO) juega el papel que antes jugaba (P_1^*W). Se verá entonces en este capítulo un posible algoritmo para resolver (W-LASSO) de forma eficiente, conocido como Weighted-FISTA (W-FISTA).

9.1. Método Gradiente Proximal

Veamos primero el método denominado “Gradiente Proximal” (GP), a partir del cual se obtendrá el algoritmo W-FISTA. Se considera para esto un problema de optimización con la siguiente estructura:

$$\begin{aligned} \operatorname{argmín} \quad & f(x) + g(x). \\ x \in \mathbb{R}^n \end{aligned} \tag{9.1}$$

Donde f es convexa y diferenciable y g es convexa y continua, no necesariamente diferenciable. El método GP combina un descenso por gradiente de la función diferenciable f , con el uso de un operador “proximal” de la función no diferenciable g . Más específicamente, utilizando un descenso con paso fijo $\alpha > 0$, el algoritmo construye la siguiente iteración [57][Sección 4.2]:

$$x^{k+1} = \operatorname{prox}_{\alpha g}(x^k - \alpha \nabla f(x^k)). \tag{9.2}$$

Donde $\operatorname{prox}_{\alpha h}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denota el operador proximal de una función $h : \mathbb{R}^n \rightarrow \mathbb{R}$. Este último se define mediante el siguiente problema de optimización estrictamente convexo [57][Sección 1.1]:

$$\operatorname{prox}_{\alpha h}(v) := \operatorname{argmín}_{x \in \mathbb{R}^n} \left\{ h(x) + \frac{1}{2\alpha} \|x - v\|_2^2 \right\}.$$

Los algoritmos GP, definidos mediante la iteración (9.2), son de tipo punto fijo. Se puede probar que \hat{x} es solución de (9.1) si y sólo si es punto fijo de (9.2) [57][Sección 4.2.1]. A su vez, si bien no siempre

el operador proximal es una contracción, resulta ser un operador “firmemente no expansivo”¹. Como se verá más adelante, esto hace que, bajo ciertas condiciones muy genéricas, la iteración GP sea convergente a un punto fijo, que por lo tanto será solución de (9.1).

El operador proximal es aplicado en cada paso de la iteración de GP. Por esta razón, para que el uso de este algoritmo tenga sentido, el cálculo del operador proximal debe ser computacionalmente sencillo. Veremos que este es el caso para el problema de interés (W-LASSO).

9.1.1. Gradiente Proximal Acelerado (GPA)

La convergencia del método GP puede ser acelerada mediante el cálculo de un punto intermedio:

$$y^k = x^k + r_k(x^k - x^{k-1}), \forall k \geq 1, r_k \in [0, 1).$$

Luego se actualiza la estimación mediante GP, pero aplicado a este punto intermedio:

$$x^{k+1} = \text{prox}_{\alpha g}(y^k - \alpha \nabla f(y^k)), \forall k \geq 1.$$

Una forma sencilla de tomar r_k es [57][Sección 4.2]:

$$r_k = \frac{k}{k+3}, \forall k \geq 1.$$

El algoritmo requiere dos puntos iniciales: x^0 y x^1 . Si se inicia con $x^1 = x^0$, el primer punto estimado corresponde a un paso de GP sin acelerar:

$$x^2 = \text{prox}_{\alpha g}(x^1 - \alpha \nabla f(x^1)).$$

9.1.2. Convergencia de GPA

Se tiene el siguiente resultado de convergencia en valor funcional para el algoritmo GPA.

Teorema 15. ([57][Sección 4.3]) *Consideremos el problema (9.1) para el caso en que ∇f es Lipschitz continua de constante $L > 0$. Si se toma $\alpha \in (0, \frac{1}{L}]$, el algoritmo GPA converge al valor funcional óptimo. En el peor caso el orden de convergencia en valor es $O(\frac{1}{k^2})$. Es decir que existe $C > 0$ con:*

$$(f + g)(x^k) - (f + g)^* \leq \frac{C}{k^2}.$$

A modo de referencia, bajo las mismas hipótesis, el algoritmo GP converge en valor funcional con orden sub-lineal: $O(\frac{1}{k})$. El orden de convergencia obtenido con GPA es óptimo, en el sentido de que

¹Un operador $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ se dice firmemente no expansivo (“firmly nonexpansive”) si:

$$\|T(x) - T(y)\|_2^2 \leq \langle T(x) - T(y), x - y \rangle, \forall x, y \in \mathbb{R}^n.$$

Utilizando la desigualdad de Cauchy-Schwarz se puede ver fácilmente que estos son un caso particular de operador Lipschitz continuo con constante unidad.

no puede ser mejorado por ningún “método de primer orden”; donde se denomina método de primer orden a los que se basan únicamente en el gradiente (o diferencial de primer orden).

9.2. Algoritmo W-FISTA para el problema W-LASSO

Aplicado al problema (W-LASSO), el algoritmo Gradiente Proximal se conoce como W-ISTA (Weighted Iterative Shrinkage-Thresholding Algorithm). En este caso la parte no diferenciable de la función objetivo de (W-LASSO) es:

$$g(x) = \|Wx\|_1 = \sum_{i=1}^n w_i |x_i| = \sum_{i=1}^n g_i(x_i) \quad \text{con} \quad g_i(x_i) = w_i |x_i|.$$

Por lo tanto, utilizando la propiedad conocida como “separabilidad de la suma”, se obtiene una expresión para la función coordenada i -ésima del operador proximal, de parámetro $\alpha > 0$:

$$(prox_{\alpha g}(x))_i = (prox_{\alpha \|Wx\|_1}(x))_i = prox_{\alpha g_i}(x_i) = \begin{cases} x_i - \alpha w_i & x_i \geq \alpha w_i \\ 0 & |x_i| \leq \alpha w_i \\ x_i + \alpha w_i & x_i \leq -\alpha w_i \end{cases}.$$

Este tipo de función se conoce como Soft-Thresholding. En la Figura 9.1 se muestra su gráfico para el caso $\tau = \frac{1}{2}$, $w_i = 3$. En el caso general el umbral se da en τw_i . Una forma computacionalmente eficiente de hallar el operador proximal es:

$$(prox_{\alpha g}(v))_i = \max(|v_i| - \alpha w_i, 0) \text{signo}(v_i).$$



Figura 9.1: Función Soft-Thresholding para $\alpha = \frac{1}{2}$ y pesos $w_i = 3$.

Por otro lado, la parte diferenciable de la función objetivo de (W-LASSO) es:

$$f(x) = \frac{\lambda}{2} \|\Phi x - b\|_2^2,$$

y por lo tanto:

$$\nabla f(x) = \lambda \Phi^T (\Phi x - b).$$

La versión acelerada de W-ISTA, la cual se puede obtener aplicando el algoritmo GPA al problema W-LASSO, es conocida como W-FISTA, y fue introducida por Beck y Teboulle en [6] para el caso sin pesos. Para esto se basaron en el método de aceleración presentado por Nesterov para funciones diferenciables [56].

9.2.1. Convergencia de W-FISTA

Veamos que ∇f es Lipschitz continua:

$$\|\nabla f(x) - \nabla f(y)\|_2 = \|\lambda \Phi^T \Phi x - \lambda \Phi^T \Phi y\|_2 = \|\lambda \Phi^T \Phi (x - y)\|_2 \leq \lambda \|\Phi^T \Phi\|_{M2} \|x - y\|_2.$$

Una constante Lipschitz es entonces: $L = \lambda \|\Phi^T \Phi\|_{M2}$; donde $\|\cdot\|_{M2}$ denota la norma 2 matricial, inducida por la respectiva norma 2 vectorial:

$$\|B\|_{M2} := \sup_{x \neq \vec{0}} \frac{\|Bx\|_2}{\|x\|_2} = \max_i \{\sigma_i(B)\}; \text{ siendo } \sigma_i(B) \text{ el } i\text{-ésimo valor singular de } B.$$

Por lo tanto, usando el Teorema 15, se deduce que el algoritmo W-FISTA resulta convergente en valor funcional, con orden $O\left(\frac{1}{k^2}\right)$, siempre que se utilice $\alpha \in \left(0, \frac{1}{\lambda \|\Phi^T \Phi\|_{M2}}\right]$.

9.2.2. Condición de parada

Como condición de parada para el algoritmo W-FISTA, se utiliza el error relativo entre las dos últimas iteraciones, evitando dividir por cero:

$$e^{k+1} := \begin{cases} 0 & \text{si } \|x^{k+1} - x^k\|_2 = 0 \\ \frac{\|x^{k+1} - x^k\|_2}{\max(\|x^{k+1}\|_2, \|x^k\|_2)} & \text{otro caso} \end{cases} \quad \forall k \geq 0. \quad (9.3)$$

En el Algoritmo (7) se muestra un pseudocódigo del método W-FISTA, utilizado para resolver el problema (W-LASSO).

Algorithm 7 Algoritmo W-FISTA para resolver (W-LASSO)

Require: $\Phi \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $w^0 \geq \vec{0}$, $x^0 \in \mathbb{R}^n$, $\alpha > 0$, $\text{tol} \geq 0$, $\text{maxIter} \geq 1$

1:
2: $x^1 = x^0$
3:
4: $e^0 = 10\text{tol}$
5: $k = 1$
6: **while** $k \leq \text{maxIter}$ **and** $e^k > \text{tol}$ **do**
7: $r_k = \frac{k}{k+3}$
8:
9: $y^k = x^k + r_k (x^k - x^{k-1})$ {punto intermedio}
10:
11: $x^{k+1} = \text{prox}_{\alpha g} (y^k - \alpha \nabla f(y^k))$ {gradiente proximal}
12:
13: calcular e^{k+1} mediante (9.3)
14:
15: $k = k + 1$
16: **end while**
17: **return** x^k

Capítulo 10

Resultados experimentales (sin ruido)

En este capítulo se analiza el desempeño del algoritmo $RW\ell_1$ de CWB [20], y de los algoritmos $RW\ell_1$ y RW-LASSO propuestos en este trabajo, los cuales actualizan los pesos w^k mediante subgradiente proyectado. Se considera únicamente el caso en que las medidas no están afectadas por ruido.

10.1. Modelo utilizado

Para cada una de las pruebas se considera un sistema lineal $\Phi x = b$ aleatorio, que admite al menos una solución $x^* \in \Sigma_k$. La forma de generar el problema se basa en [20][Sección 3.1]:

1. Se genera $\Phi \in \mathbb{R}^{m \times n}$, con $n = 256$, $m < n$ y entradas aleatorias gaussianas e independientes:

$$\Phi_{ij} \sim N\left(0, \sigma = \frac{1}{\sqrt{m}}\right), \forall i, j.$$

De esta forma Φ cumple con las hipótesis del Teorema 8, garantizando $RIP(2k)$ con $\delta_{2k} < \sqrt{2}-1$, con alta probabilidad. Recordar que esta forma de generar Φ implica que sus columnas tendrán norma 2 unidad (en valor esperado), por lo que no es necesario normalizar sus columnas.

2. Se sortea sin repetición un conjunto de k índices $I_k \subset \{1, \dots, n\}$, que representa las posiciones no nulas de la solución buscada x^* . Es decir que I_k es el soporte de x^* .
3. Se generan los valores no nulos de x^* con coordenadas independientes gaussianas:

$$x_i^* \sim N\left(0, \sigma = \frac{1}{\sqrt{k}}\right), \forall i \in I_k.$$

De esta forma x^* tendrá norma 2 unidad en valor esperado, al igual que las columnas de Φ .

4. Se genera el término independiente: $b = \Phi x^* \in \mathbb{R}^m$.

Al momento de realizar las pruebas los datos disponibles son: Φ y b . Se busca recuperar x^* .

10.2. Valores iniciales y parámetros

En todos los algoritmos RW se utiliza:

- Pesos iniciales: $w^0 = \vec{1}$.
- Estimación inicial: solución de $\Phi x = b$ de norma 2 mínima. Si se asume Φ de rango máximo, esta viene dada por:

$$x^0 = \Phi^+ b = \Phi^T (\Phi \Phi^T)^{-1} b.$$

- “Warm-Restart”: en el paso k de cada algoritmo RW, en el cual se debe calcular x^{k+1} , se toma la solución del paso anterior como condición inicial: $x^0 = x^k$.

10.3. Evaluación del desempeño

Siguiendo el criterio propuesto en [20][Sección 3.1], en cada prueba se considerará que el algoritmo logra recuperar x^* si se cumple:

$$\|x^{\text{maxIter}} - x^*\|_\infty \leq 1 \times 10^{-3}.$$

En las simulaciones de este capítulo no se considera el algoritmo RW-LASSO subgradiente pues, con el criterio antes definido, su porcentaje de recuperación resulta siempre nulo. Posiblemente esto se deba a que las soluciones del problema W-LASSO no necesariamente deben cumplir $\Phi x = b$, contrario a lo que ocurre en los problemas Weighted ℓ_1 sin ruido.

10.4. Cantidad de medidas m fija

Se considera inicialmente el caso en que se dispone de una cantidad fija m de medidas y se desea analizar el desempeño del algoritmo al aumentar k , que en este caso denota la pseudo-norma 0 de la solución buscada x^* . En particular se toma $m = 100$ y se realizan pruebas con $k \in [15, 55]$. Para cada k se generan N_p problemas aleatorios y se mide la cantidad de veces que el algoritmo logra recuperar x^* .

10.4.1. Elección del parámetro ϵ

El parámetro $\epsilon > 0$ de los algoritmos RW se selecciona de forma experimental, realizando $N_p = 100$ pruebas con 4 iteraciones de RW cada una, y para distintos valores de ϵ . En la Figura 10.1 se muestran los resultados obtenidos para el algoritmo $\text{RW}\ell_1$ de CWB. El mejor comportamiento se da en este caso con $\epsilon = 0.1$, por lo que en las siguientes pruebas se tomará este valor para dicho algoritmo.

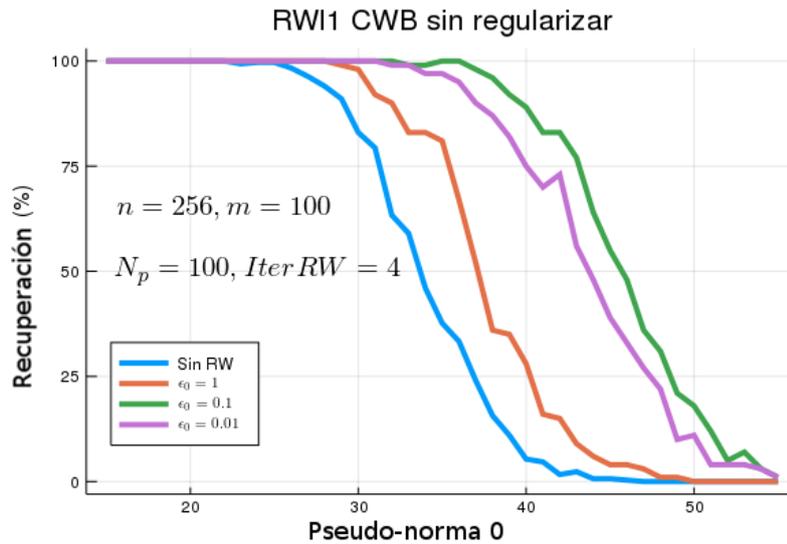


Figura 10.1: Porcentaje de recuperación según k del algoritmo $RW\ell_1$ de CWB sin regularizar. Se realizan 4 iteraciones de RW para distintos valores de ϵ . El valor con el cual se logra mejor desempeño es $\epsilon = 0.1$. Se muestra como referencia el desempeño de la minimización ℓ_1 , sin RW.

En la Figura 10.2 se muestran los resultados obtenidos para el algoritmo $RW\ell_1$ subgradiente. Los porcentajes de recuperación son idénticos para los tres valores de ϵ . Esto es coherente con el análisis teórico del algoritmo sin oráculo, en el que se concluyó que el valor de ϵ resulta irrelevante, siempre que sea positivo. En lo que sigue se utilizará $\epsilon = 1$ para este algoritmo.

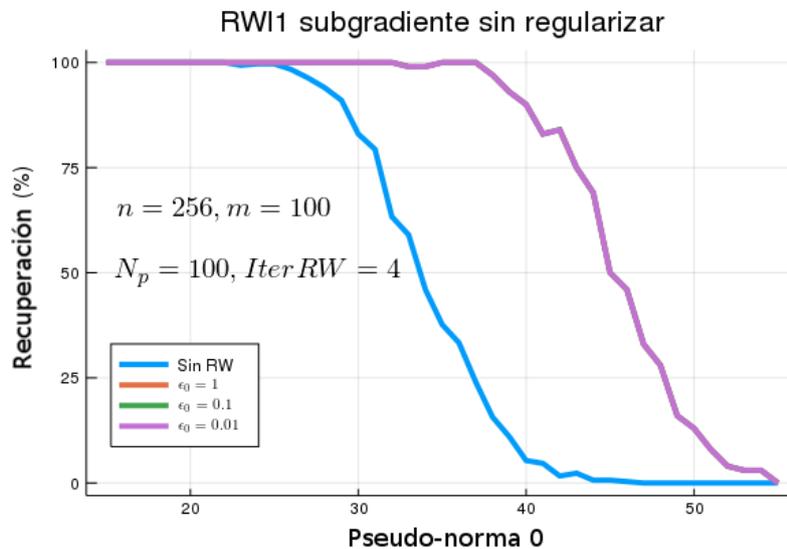


Figura 10.2: Porcentaje de recuperación según k del algoritmo $RW\ell_1$ con subgradiente proyectado y sin regularizar. Se realizan 4 iteraciones de RW para distintos valores de ϵ . En todos los casos el desempeño es idéntico. Se muestra como referencia el desempeño de la minimización ℓ_1 , sin RW.

El hecho de que el algoritmo $RW\ell_1$ propuesto no dependa del valor del parámetro ϵ significa una ventaja respecto al $RW\ell_1$ de CWB, dado que en este último resulta necesario seleccionar el parámetro según las características de cada problema CS.

10.4.2. Desempeño según las iteraciones de RW

En la Figura 10.3 se muestra el porcentaje de recuperación de ambos algoritmos $RW\ell_1$ sin regularizar, y para distinta cantidad de iteraciones de RW. En este caso se realizan $N_p = 300$ pruebas por cada valor de k . Para ambos algoritmos $RW\ell_1$ el desempeño en porcentaje de recuperación es superior al de la minimización ℓ_1 . A su vez, el porcentaje de recuperación mejora al aumentar la cantidad de iteraciones de RW, aunque dicha mejora parece ser cada vez menos significativa.

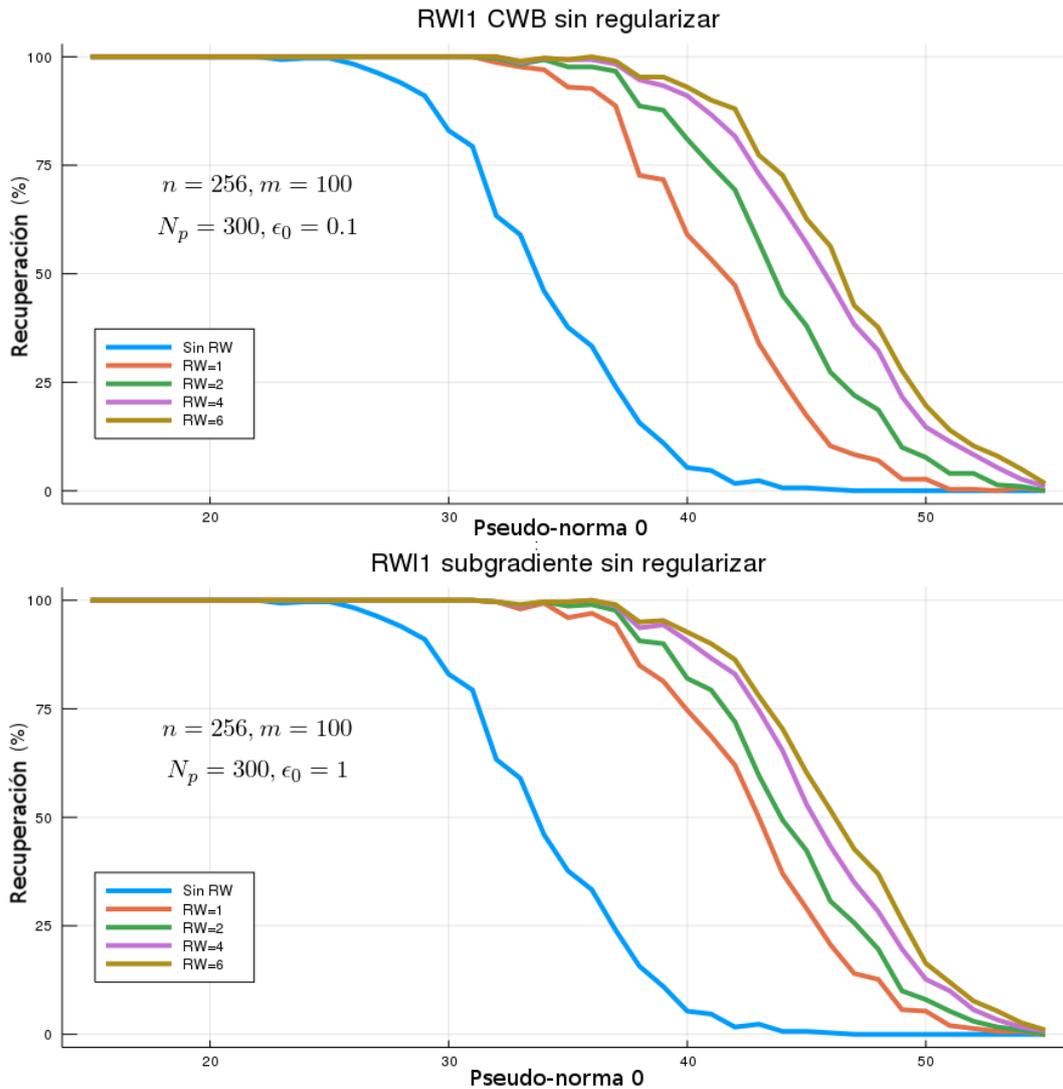


Figura 10.3: Porcentaje de recuperación de los algoritmos $RW\ell_1$ de CWB (arriba) y de subgradiente (abajo), al aumentar k , pseudo-norma 0 de la solución x^* buscada. Se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 100$ medidas lineales. En ambos casos las pruebas son sin regularizar y las medidas no presentan ruido.

Utilizando los mismos resultados presentados en la Figura 10.3, la Figura 10.4 compara el porcentaje de recuperación de cada algoritmo $RW\ell_1$, para distinta cantidad de iteraciones RW. Si solamente se realiza una iteración de RW, el porcentaje de recuperación del algoritmo $RW\ell_1$ propuesto es algo superior al de CWB. Esta diferencia resulta imperceptible a partir de dos iteraciones de RW, donde ambos algoritmos logran un desempeño comparable.

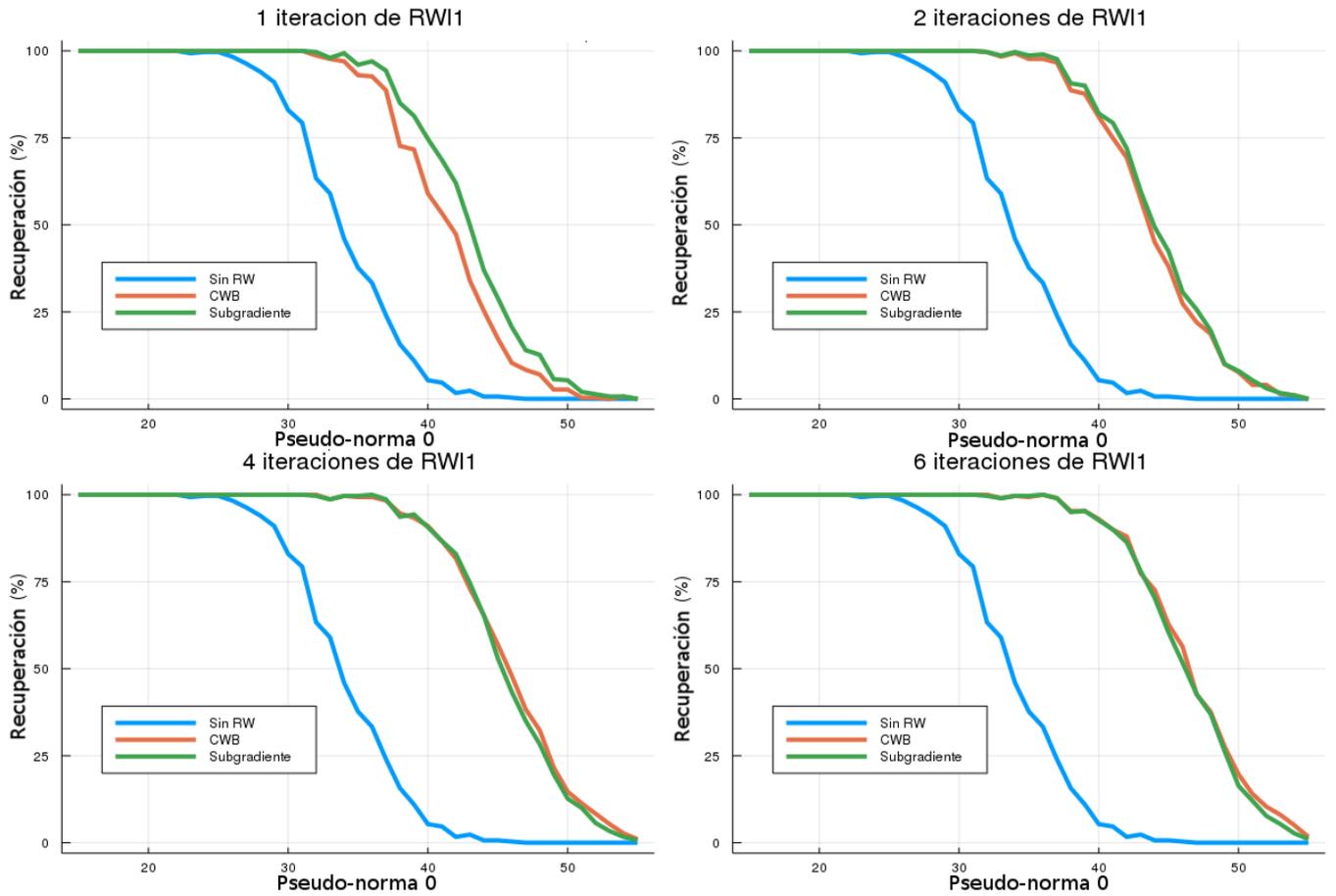


Figura 10.4: Porcentaje de recuperación de los algoritmos $RW\ell_1$ al aumentar k , pseudo-norma 0 k de la solución x^* buscada, y para distinta cantidad de iteraciones de RW. En cada caso se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 100$ medidas lineales. En todos los casos las pruebas son sin regularizar y las medidas no presentan ruido.

Notar que en todos los gráficos se aprecia una zona de transición, más o menos angosta según el algoritmo, entre los valores de k donde se tiene probabilidad uno de recuperación, y los valores donde esta probabilidad es nula. Esto corresponde al fenómeno de transición de fase, analizado en la Sección 3.2.3 para el caso de minimización ℓ_1 sin pesos.

10.5. Nivel de esparcidad k fijo

Se considera ahora el caso en que el nivel de esparcidad k está dado y se desea analizar el desempeño del algoritmo al variar la cantidad de medidas m realizadas. Se toma en particular $k = 35$ y se varía $m \in [2k, \frac{n}{2}] = [70, 128]$, con incrementos de tamaño 2. Para cada algoritmo se utilizan los mismos valores de ϵ obtenidos en la sección anterior.

10.5.1. Desempeño según las iteraciones de RW

En la Figura 10.5 se muestra el porcentaje de recuperación de ambos algoritmos $RW\ell_1$ sin regularizar, y para distinta cantidad de iteraciones de RW. Por cada valor de m se realizan $N_p = 300$ pruebas. Para ambos algoritmos $RW\ell_1$ el desempeño es superior al de la minimización ℓ_1 . A su vez, al

igual que con los resultados obtenidos al variar k , el porcentaje de recuperación mejora al aumentar la cantidad de iteraciones de RW, aunque dicha mejora parece ser cada vez menos significativa.

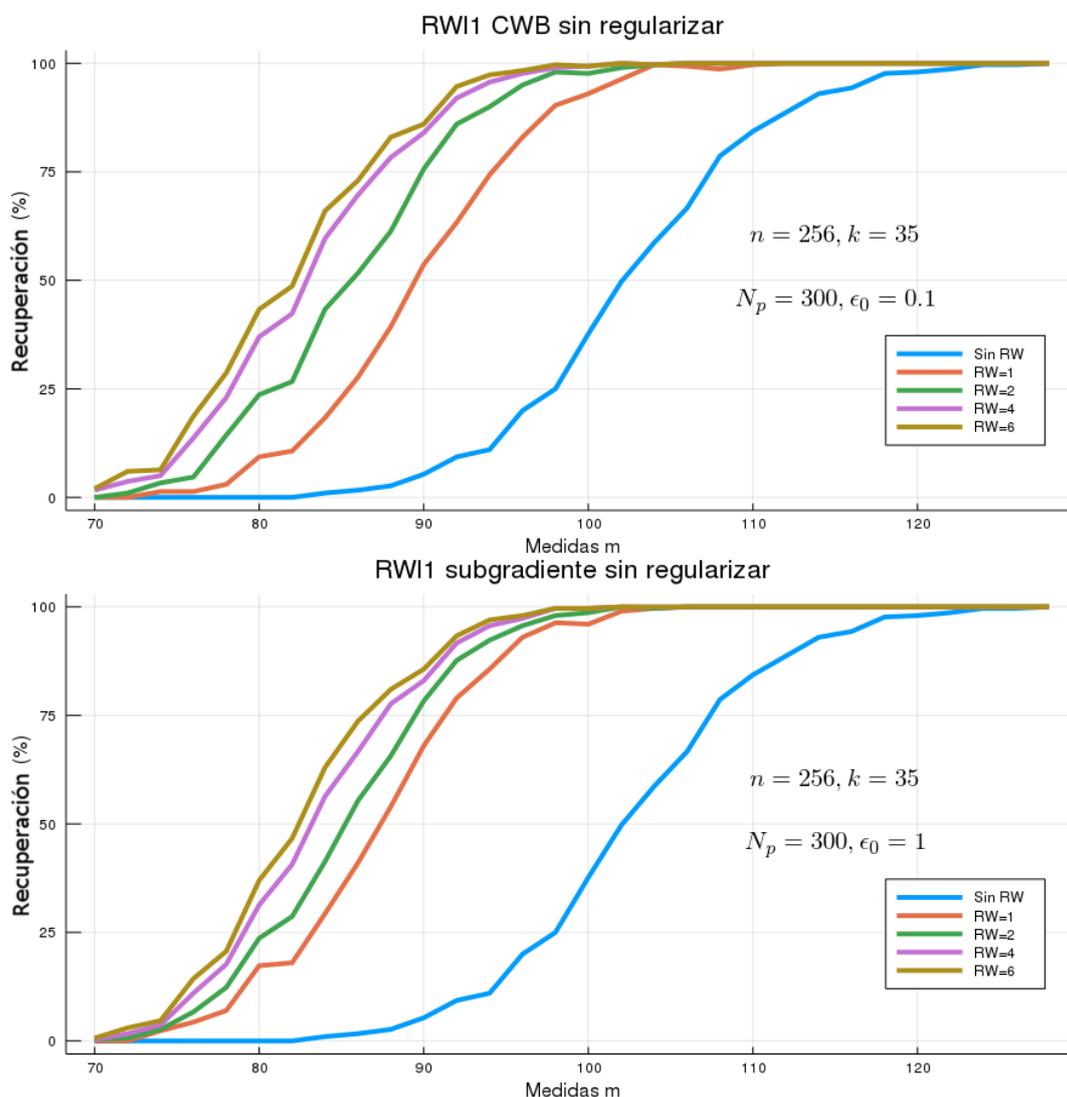


Figura 10.5: Porcentaje de recuperación de los algoritmos $\text{RW}\ell_1$ de CWB (arriba) y de subgradiente (abajo), al aumentar la cantidad de medidas m de la solución x^* buscada. Se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y pseudo-norma 0 $k = 35$. En ambos casos las pruebas son sin regularizar y las medidas no presentan ruido.

Utilizando los mismos resultados presentados en la Figura 10.5, la Figura 10.6 compara el porcentaje de recuperación de cada algoritmo $\text{RW}\ell_1$, para distinta cantidad de iteraciones RW.

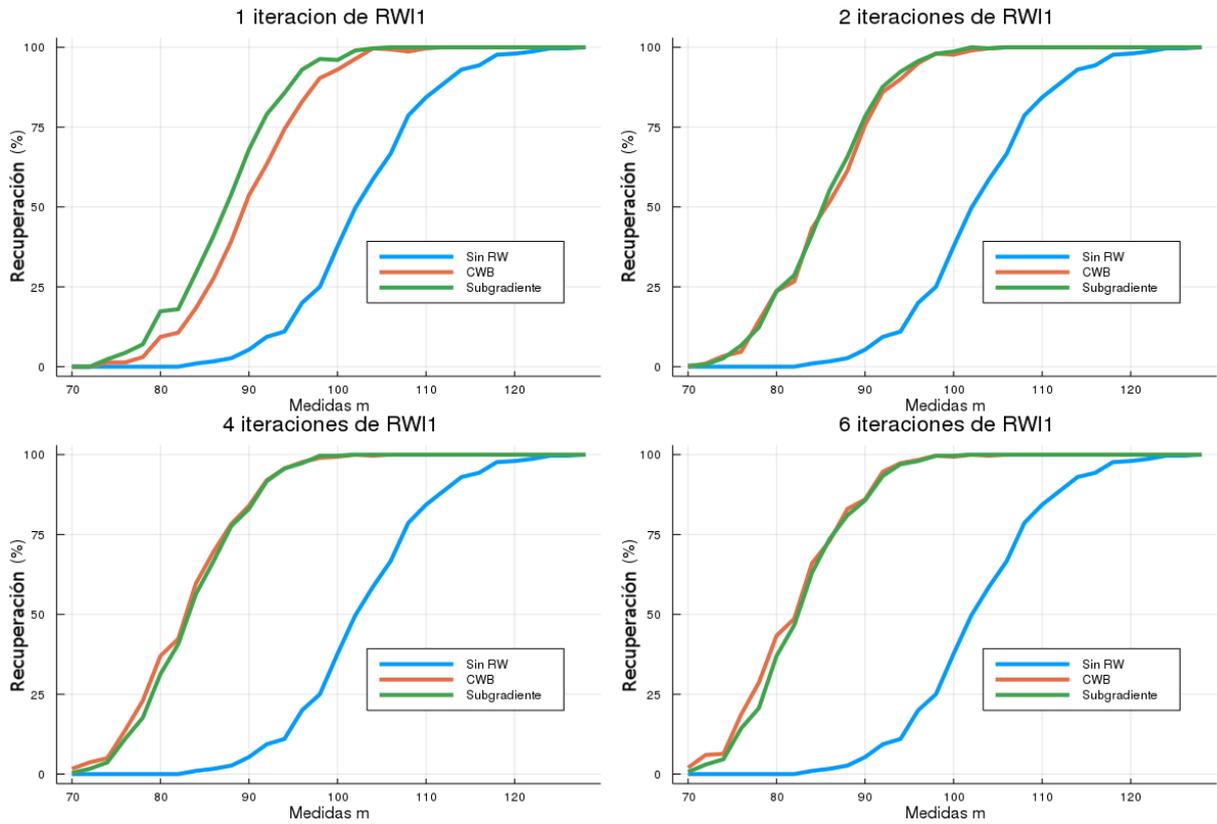


Figura 10.6: Porcentaje de recuperación de los algoritmos $RW\ell_1$ al aumentar la cantidad de medidas m de la solución x^* buscada, y para distinta cantidad de iteraciones de RW. En cada caso se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y pseudo-norma 0 $k = 35$. En todos los casos las pruebas son sin regularizar y las medidas no presentan ruido.

Cualitativamente el desempeño comparado entre ambos algoritmos es idéntico al de las pruebas realizadas variando el valor de k . Esto es: si solamente se realiza una iteración de RW, el porcentaje de recuperación del algoritmo $RW\ell_1$ propuesto es algo superior al de CWB. Esta diferencia resulta imperceptible a partir de dos iteraciones de RW, donde ambos algoritmos logran un desempeño comparable.

10.6. $RW\ell_1$ CWB regularizado

En esta sección se considera el desempeño del algoritmo $RW\ell_1$ de CWB, actualizando ϵ_k de la misma forma que en el algoritmo $RW\ell_2$ regularizado, que fuera expuesto en la Sección 4.4.2. Nos referiremos a esta implementación como $RW\ell_1$ CWB regularizado. No se analiza la regularización del algoritmo $RW\ell_1$ subgradiente (sin oráculo) pues, tal como se obtuvo de forma teórica, y luego se comprobó experimentalmente, el valor del parámetro ϵ no afecta su desempeño. Para seleccionar el valor inicial ϵ_0 del parámetro regularizador de CWB, se realizan pruebas con distintos valores de dicho parámetro, cada una con una cantidad máxima de 6 iteraciones de RW. La Figura 10.7 muestra los resultados obtenidos. Los mejores resultados se obtienen con un valor inicial $\epsilon_0 = 0.1$, al igual que en el caso sin regularizar, donde ϵ se mantenía fijo.

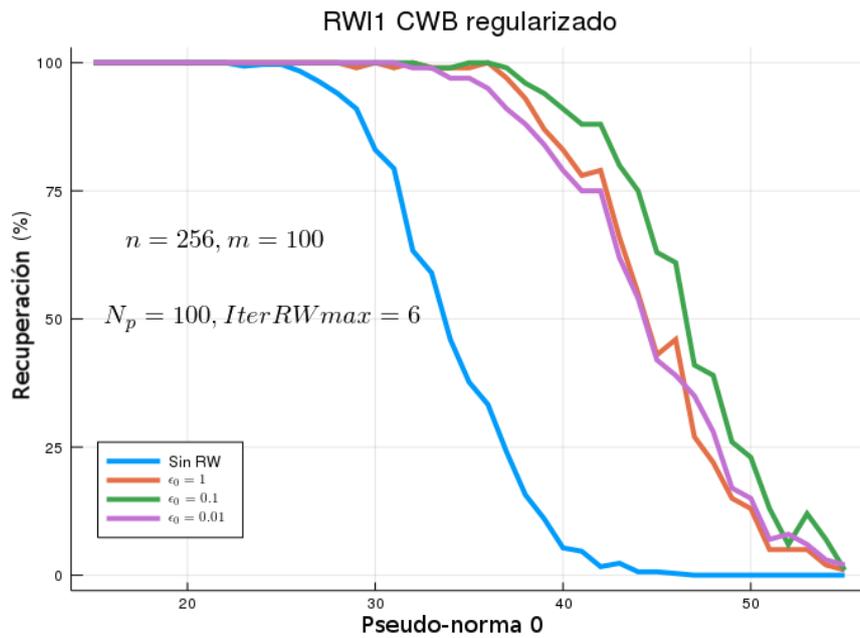


Figura 10.7: Porcentaje de recuperación del algoritmo $RW\ell_1$ de CWB regularizado, en función de la pseudo-norma 0 de la solución x^* buscada, y para distintos valores del regularizador inicial ϵ_0 . La cantidad máxima de iteraciones de RW se fija en 6. En cada caso se realizan $N_p = 100$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 100$, en el cual las medidas no presentan ruido.

En la Figura 10.8 se compara el desempeño del algoritmo $RW\ell_1$ de CWB, regularizado y sin regularizar, y para distinta cantidad de iteraciones. En ambos casos se inicia con $\epsilon_0 = 0.1$, valor que se mantiene incambiado en cada iteración del caso sin regularizar.

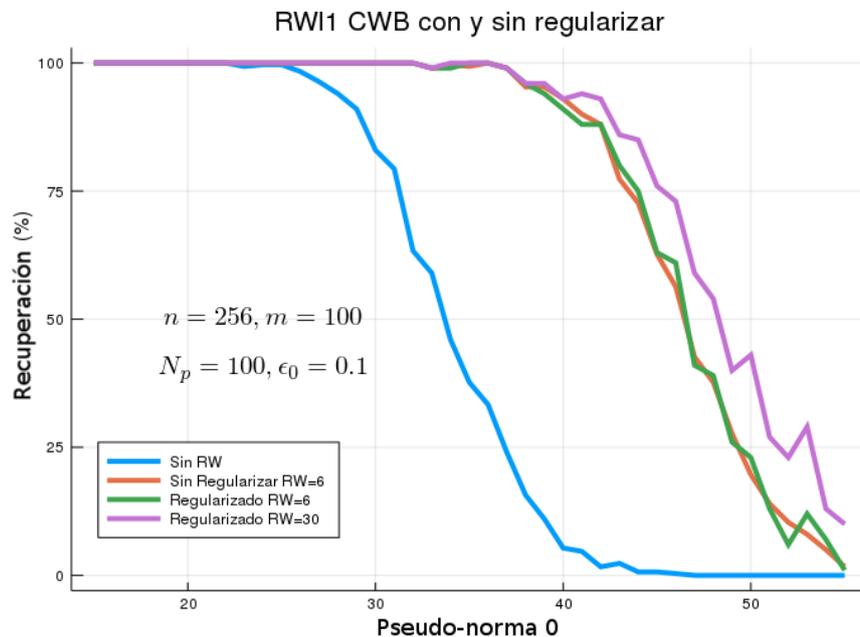


Figura 10.8: Porcentaje de recuperación del algoritmo $RW\ell_1$ de CWB regularizado y sin regularizar, en función de la pseudo-norma 0 de la solución x^* buscada, y para distinta cantidad de iteraciones RW. En cada caso se realizan $N_p = 100$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 100$, en el cual las medidas no presentan ruido.

Si se admiten hasta 6 iteraciones de RW, no se obtienen diferencias entre el desempeño del algoritmo regularizado y sin regularizar. Esto puede deberse a que el efecto de la regularización

requiera de mayor cantidad de iteraciones para ser efectivo. Por otro lado, para 30 iteraciones máximas de RW, el desempeño del algoritmo regularizado, si bien es superior al caso de 6 iteraciones, no introduce una mejora tal que amerite triplicar el costo computacional.

Capítulo 11

Resultados experimentales (con ruido Gaussiano)

En este capítulo se analiza el desempeño del algoritmo $\text{RW}\ell_1$ de CWB [20], y de los algoritmos $\text{RW}\ell_1$ y RW-LASSO propuestos en este trabajo, los cuales actualizan los pesos w^k mediante subgradiente proyectado. Se considera el caso en que las medidas están afectadas por ruido aditivo gaussiano.

11.1. Modelo utilizado

- La matriz Φ y la solución buscada x^* se generan de igual forma que para las pruebas sin ruido. Siguiendo el criterio de [20], se utilizan $m = 128$ medidas en lugar de $m = 100$. Al igual que antes se toma $n = 256$.
- El vector de medidas es: $b = \Phi x^* + z$, donde z representa el ruido.
- El ruido se genera con coordenadas Gaussianas e independientes $z_i \sim N(0, \sigma^2)$, y de forma que x^* sea factible con alta probabilidad: $\|\Phi x^* - b\|_2 \leq \eta$. Para esto se toma:

$$z_i = \sigma v_i, \quad v_i \sim N(0, 1) \text{ independientes.}$$

Entonces:

$$\|v\|_2^2 = \sum_{i=1}^m v_i^2 \sim \chi_m^2 \quad \text{chi-cuadrado, de } m \text{ grados de libertad.}$$

Por lo tanto $\|v\|_2^2$ tendrá media m y desviación estándar $\sqrt{2m}$. Luego, tomando por ejemplo $\eta^2 = \sigma^2 (m + 2\sqrt{2m})$, y recordando que $z = \sigma v$, se tiene:

$$P(\|\Phi x^* - b\|_2^2 \geq \eta^2) = P(\|z\|_2^2 \geq \sigma^2 (m + 2\sqrt{2m})) = P(\|v\|_2^2 \geq m + 2\sqrt{2m}).$$

En particular para $m = 128$, a partir de la tabla de valores de la distribución χ_{128}^2 , se obtiene que la probabilidad de que x^* sea factible es:

$$P(\|\Phi x^* - b\|_2^2 \leq \eta^2) = 1 - P(\|\Phi x^* - b\|_2^2 \geq \eta^2) = 1 - P(\chi_{128}^2 \geq 160) \simeq 1 - 0.0291 \simeq 0.971.$$

El valor específico de σ se selecciona según [20][Sección 3.3]:

$$\sigma = \beta \frac{\|\Phi x^*\|_2}{\|v\|_2}, \quad \beta = 0.2.$$

11.2. Valores iniciales y parámetros

Se utilizan los mismos valores iniciales y parámetros que para el caso sin ruido. Para el algoritmo RW-LASSO subgradiente se utiliza:

- Multiplicador inicial: $\lambda^0 = \frac{n}{\|x^0\|_1}$.
- Iteraciones máximas W-FISTA: 500.
- Tolerancia de parada W-FISTA: 1×10^{-3} .

11.3. Evaluación del desempeño

En lugar de medir el porcentaje de recuperación de x^* , se calcula la posible mejora respecto a una solución $x_{\ell_1}^\eta$, obtenida al minimizar $(P_1^\eta W)$ con pesos unidad (minimización ℓ_1 con ruido). Esto es:

$$a = 100 \times \left(1 - \frac{\|x^{RW} - x^*\|_2}{\|x_{\ell_1}^\eta - x^*\|_2} \right) \%.$$

Valores de $a > 0$ indican una mejora del algoritmo $RW\ell_1$ respecto a la minimización ℓ_1 . En ese caso la mejora es del a por ciento.

11.4. Resultados obtenidos

Siguiendo el criterio de [20][Sección 3.3], se realizan $N_p = 300$ pruebas con esparcidad $k = 38$ y un máximo de $\text{maxIter} = 9$ iteraciones de RW. Inicialmente se analiza cómo afecta el parámetro ϵ de las restricciones no ideales al desempeño del algoritmo RW-LASSO, así como a su versión regularizada. La Figura 11.1 muestra los resultados obtenidos para distintos valores de ϵ_0 inicial, con y sin regularizar. En todos los casos la mejora promedio respecto a la minimización ℓ_1 es del orden de 31 %, y la desviación estándar del 10 al 12 %. En particular estos valores no varían considerablemente para los distintos valores de ϵ_0 , ni entre el caso con y sin regularizar. Esto último puede deberse a que se requiera mayor cantidad de iteraciones para notar el efecto de la regularización.

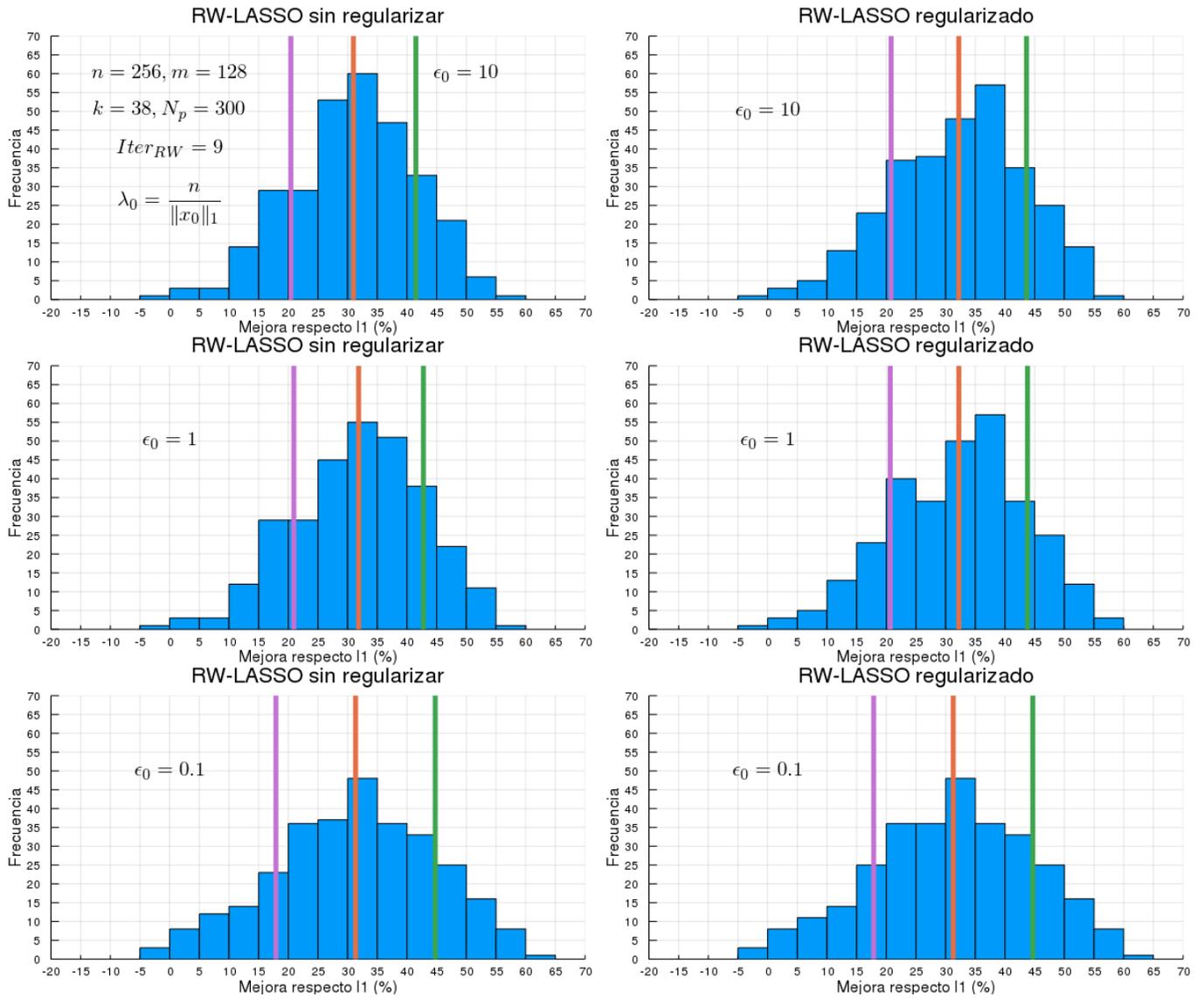


Figura 11.1: Mejora del algoritmo RW-LASSO respecto a la minimización ℓ_1 . Medidas afectadas por ruido gaussiano aditivo $N\left(0, \frac{1}{\sqrt{k}}\right)$. En cada gráfico se indica la media (línea roja vertical) del porcentaje de mejora a , \pm una desviación estándar (líneas verticales violeta y verde). En cada caso se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 128$ medidas lineales con ruido.

El hecho de que se observen diferencias al variar el parámetro ϵ en RW-LASSO subgradiente es coherente con el análisis teórico del algoritmo, en el cual se obtuvo que el valor de dicho parámetro resulta relevante, al contrario de lo que ocurriría con $RW\ell_1$ subgradiente.

Se comparan ahora los tres algoritmos RW entre sí, seleccionando los valores de ϵ de los algoritmos $RW\ell_1$ de acuerdo a los resultados sin ruido, y tomando $\epsilon = 1$ para el algoritmo RW-LASSO. La Figura 11.2 muestra los histogramas de mejora obtenidos.

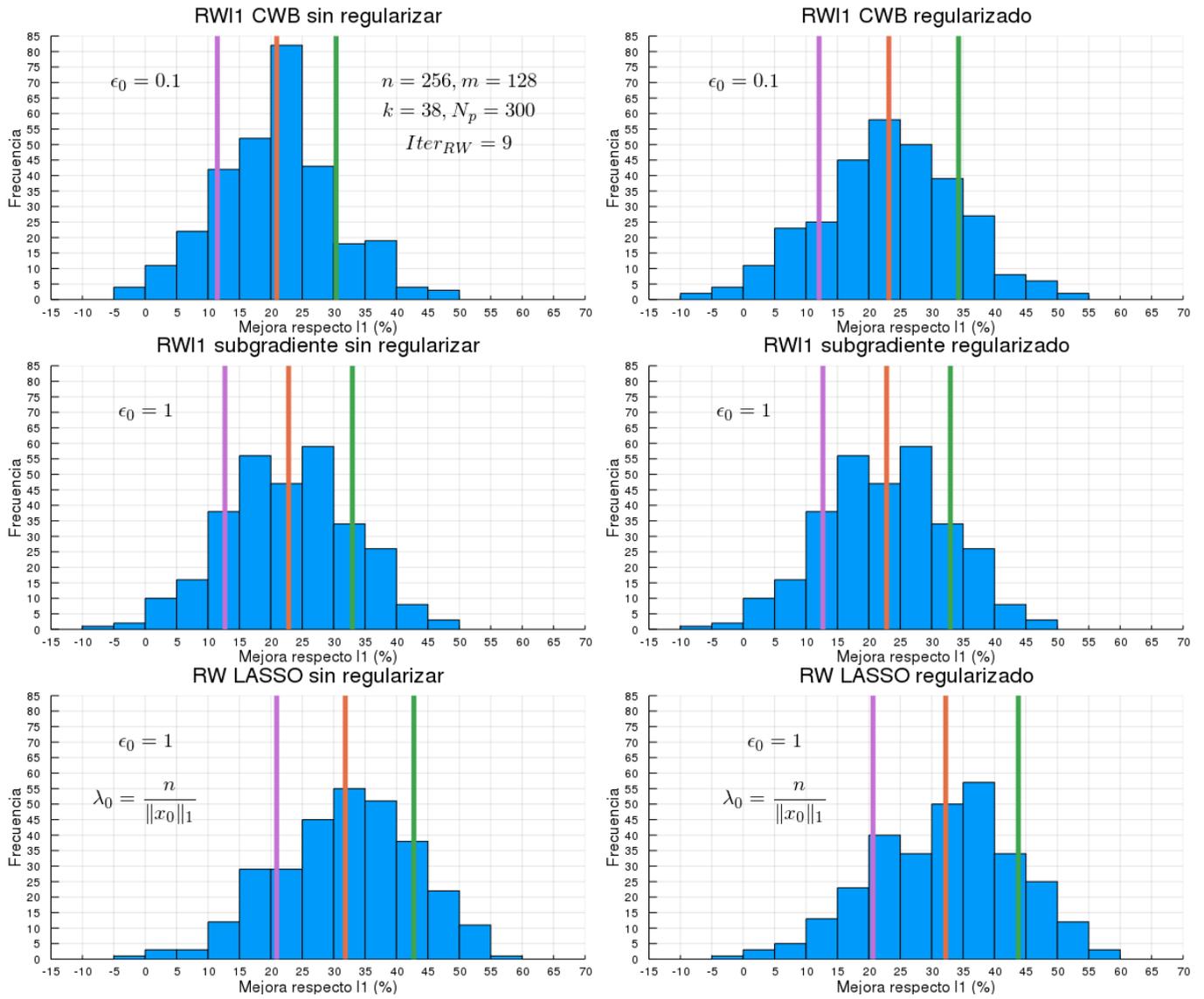


Figura 11.2: Mejora de los algoritmos RW respecto a la minimización ℓ_1 . Medidas afectadas por ruido gaussiano aditivo $N\left(0, \frac{1}{\sqrt{k}}\right)$. En cada gráfico se indica la media (línea roja vertical) del porcentaje de mejora a , \pm una desviación estándar (líneas verticales violeta y verde). En cada caso se realizan $N_p = 300$ pruebas, cada una con un problema CS aleatorio de tamaño $n = 256$ y $m = 128$ medidas lineales con ruido.

En el algoritmo $RW\ell_1$ de CWB el desempeño mejora levemente al incorporar la regularización; pasando de un par media, desviación estándar de (20.9 %, 9.46 %) a (23.2 %, 11.1 %). En el algoritmo $RW\ell_1$ subgradiente no hay diferencias entre el caso regularizado y sin regularizar. Esto es coherente con el análisis teórico de dicho algoritmo, en el cual se obtuvo que el valor de $\epsilon_k > 0$ resulta irrelevante. En ambos casos el par media, desviación estándar es: (22.8 %, 10.1 %), y su desempeño resulta muy similar al de CWB regularizado. Contrario a lo que ocurría en el caso sin ruido, ahora el desempeño de RW-LASSO es claramente superior al de los algoritmos $RW\ell_1$ de CWB y de subgradiente. Su par media, desviación estándar es: (31.8 %, 10.9 %) sin regularizar y (32.2 %, 11.6 %) regularizado. En particular la mejora promedio respecto a minimización ℓ_1 aumenta en el orden de un 40 %, pasando de 23 % en los algoritmos $RW\ell_1$ a 32 % en RW-LASSO; mientras que la desviación estándar se mantiene en el orden del 10 al 11 % para todos los algoritmos.

Capítulo 12

Conclusiones finales

En este trabajo se presentó el problema de Compressed Sensing (CS), junto con algunos de los resultados teóricos más relevantes sobre el mismo. El área de investigación vinculada a CS surge como tal a mediados de la primer década del 2000, con los trabajos de Donoho y de Candès, Romberg y Tao. Desde entonces ha habido un enorme interés en el desarrollo de la misma, motivado principalmente por sus importantes aplicaciones, algunas de las cuales se expusieron brevemente al inicio del documento.

Dado que CS es en general un problema NP-difícil, es necesario buscar problemas alternativos, de más fácil resolución, y que bajo ciertas hipótesis sean equivalentes. Una de las alternativas más usuales consiste en reemplazar el problema CS por un problema convexo, en el cual se minimiza la norma ℓ_1 . Se presentaron resultados clásicos, basados en la propiedad RIP, que brindan condiciones suficientes para garantizar la equivalencia de este último problema con CS. Como también fuera expuesto en este trabajo, la propiedad RIP permite además garantizar un buen desempeño del problema ℓ_1 ante medidas afectadas por ruido, o cuando la señal buscada no es exactamente esparsa. En particular se analizó el uso de matrices de medida Φ aleatorias, con distribución Gaussiana, para las cuales la propiedad RIP se cumple con alta probabilidad.

Más adelante se presentó el problema Weighted ℓ_1 , consistente en asignar pesos a las coordenadas de la norma ℓ_1 . Junto con este se introdujo la clase de algoritmos Re-Weighted ℓ_1 , que buscan elegir los pesos de forma de obtener una solución de CS, cuando la minimización ℓ_1 no lo logra. Se introdujo en particular el algoritmo $\text{RW}\ell_1$ de Candès, Wakin y Boyd (CWB), donde los pesos son actualizados en función de los valores obtenidos en su iteración anterior. Se presentó luego una nueva metodología para actualizar los pesos del algoritmo Re-Weighted ℓ_1 , basada en la identificación de los mismos con multiplicadores de Lagrange de un problema convexo. A partir de esta metodología, y utilizando un algoritmo estándar para hallar soluciones de un problema dual, conocido como “método subgradiente proyectado”, se propuso un nuevo algoritmo $\text{RW}\ell_1$, alternativo al presentado por CWB. Utilizando la misma metodología, junto con el método subgradiente proyectado, se propuso a su vez un algoritmo RW-LASSO , basado en el problema Weighted LASSO. Para resolver este último problema se describió e implementó el algoritmo Weighted Fast ISTA, el cual trata la no diferenciablez de la norma ℓ_1 mediante el operador conocido como “proximal”.

El desempeño de los tres algoritmos RW fue analizado mediante su aplicación a un problema

de CS aleatorio, con matriz de medida Φ Gaussiana y medidas b con y sin ruido. A partir de estas simulaciones se pudo comprobar la mejora en el desempeño de los algoritmos RW, en comparación con la minimización ℓ_1 . En particular, para una cantidad fija m de medidas lineales, se logró mejorar el porcentaje de recuperación al aumentar la pseudo-norma 0 de la solución buscada. De forma similar, para un nivel de esparcidad k fijo, se logró aumentar el porcentaje de recuperación al variar la cantidad de medidas m utilizadas, en relación a la minimización ℓ_1 . En base a estas simulaciones se comparó también el desempeño de los algoritmos RW entre sí. Para el caso de medidas sin ruido, el algoritmo $\text{RW}\ell_1$ subgradiente obtiene un desempeño muy similar al de CWB; mientras que el RW-LASSO no logra un buen desempeño. Para el caso de medidas con ruido aditivo Gaussiano, los algoritmos $\text{RW}\ell_1$ de CWB y subgradiente obtienen nuevamente un desempeño similar, logrando una mejora promedio de un 23% respecto a la minimización ℓ_1 . Por otro lado el algoritmo RW-LASSO propuesto permite obtener una mejora promedio de un 32% respecto a la minimización ℓ_1 , lo cual representa un desempeño promedio 40% superior al de los algoritmos $\text{RW}\ell_1$ analizados, y en particular al de CWB.

Referencias bibliográficas

- [1] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.
- [3] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Compressed channel sensing: A new approach to estimating sparse multipath channels. *Proceedings of the IEEE*, 98(6):1058–1076, 2010.
- [4] R. Baraniuk and P. Steeghs. Compressive radar imaging. In *2007 IEEE Radar Conference*, pages 128–133. IEEE, 2007.
- [5] R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9(1):51–77, 2009.
- [6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [7] D. Bertsekas. *Nonlinear programming*. Athena Scientific optimization and computation series. Athena Scientific, 1999. ISBN 9781886529007.
- [8] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
- [9] P. Blomgren, T. Chan, and T. Color. *Total variation methods for restoration of vector valued images*. PhD thesis, UCLA, 1998.
- [10] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.
- [11] J. Bourgain, S. Dilworth, K. Ford, S. Konyagin, D. Kutzarova, et al. Explicit constructions of rip matrices and related problems. *Duke Mathematical Journal*, 159(1):145–185, 2011.
- [12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [13] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

- [14] T. T. Cai and A. Zhang. Sharp rip bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.
- [15] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.
- [16] E. Candès and J. Romberg. ℓ_1 -magic: Recovery of sparse signals via convex programming. www.acm.caltech.edu/l1magic/downloads/l1magic.pdf, 4:14, 2005.
- [17] E. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *arXiv preprint math/0410542*, 2004.
- [18] E. Candès and T. Tao. Decoding by linear programming. *arXiv preprint math/0502327*, 2005.
- [19] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathématique*, 346(9-10):589–592, 2008.
- [20] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.
- [21] W. L. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, and D. M. Mittleman. A single-pixel terahertz imaging system based on compressed sensing. *Applied Physics Letters*, 93(12):121105, 2008.
- [22] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3869–3872. IEEE, 2008.
- [23] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.
- [24] J. F. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38(5):826–844, 1973.
- [25] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.
- [26] M. A. Davenport. *Random observations on random observations: Sparse signal acquisition and processing*. PhD thesis, Rice University, 2010.
- [27] M. Davies and R. Gribonval. Restricted isometry constants where ℓ_p sparse recovery can fail for $0 < p \leq 1$. 2008.
- [28] A. Domahidi, E. Chu, and S. Boyd. Ecos: An socp solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076. IEEE, 2013.

- [29] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.
- [30] D. L. Donoho and B. F. Logan. Signal recovery and the large sieve. *SIAM Journal on Applied Mathematics*, 52(2):577–591, 1992.
- [31] D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.
- [32] D. L. Donoho and J. Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98(6):913–924, 2010.
- [33] D. L. Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
- [34] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.
- [35] Y. C. Eldar and G. Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
- [36] J. Ellenberg. Fill in the blanks. https://www.wired.com/2010/02/ff_algorithm/, 2010. Consultado: 30-01-2019.
- [37] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.
- [38] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013. ISBN 978-0-8176-4947-0.
- [39] Y. Gao and M. Ma. A new bound on the block restricted isometry constant in compressed sensing. *Journal of inequalities and applications*, 2017(1):174, 2017.
- [40] B. Hayes. The best bits. *American scientist*, 97(4):276–280, 2009. URL <https://www.americanscientist.org/article/the-best-bits>.
- [41] G. Hennenfent and F. J. Herrmann. Simply denoise: Wavefield reconstruction via jittered undersampling. *Geophysics*, 73(3):V19–V28, 2008.
- [42] M. A. Herman and T. Strohmer. High-resolution radar via compressed sensing. *arXiv preprint arXiv:0803.2257*, 2008.
- [43] A. Izmailov and M. Solodov. *Otimização, volume 2: métodos computacionais*. IMPA, 2012.
- [44] L. Jacques and P. Vandergheynst. Compressed sensing: When sparsity meets sampling. *Chapter*, 23:507–527, 2010.

- [45] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye. k-t focuss: a general compressed sensing framework for high resolution dynamic mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(1):103–116, 2009.
- [46] G. Kutyniok. Theory and applications of compressed sensing. *GAMM-Mitteilungen*, 36(1):79–101, 2013.
- [47] S. Li, L. Da Xu, and X. Wang. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, 2013.
- [48] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 145–156. ACM, 2009.
- [49] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72, 2008.
- [50] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.
- [51] P. Nagesh and B. Li. Compressive imaging of color images. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1261–1264. IEEE, 2009.
- [52] D. Needell. Noisy signal recovery via iterative reweighted ℓ_1 -minimization. *arXiv preprint arXiv:0904.3780*, 2009.
- [53] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.
- [54] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9(3):317–334, 2009.
- [55] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.
- [56] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O\left(\frac{1}{k^2}\right)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.
- [57] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [58] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

- [59] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee. Compressive sensing: From theory to applications, a survey. *Journal of Communications and networks*, 15(5):443–456, 2013.
- [60] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [61] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.
- [62] Y. Sun, P. Babu, and D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2017.
- [63] G. Tauböck, F. Hlawatsch, D. Eiwien, and H. Rauhut. Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing. *arXiv preprint arXiv:0903.2774*, 2009.
- [64] H. L. Taylor, S. C. Banks, and J. F. McCoy. Deconvolution with the ℓ_1 norm. *Geophysics*, 44(1):39–52, 1979.
- [65] Z. Tian and G. B. Giannakis. Compressed sensing for wideband cognitive radios. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP'07*, volume 4, pages IV–1357. IEEE, 2007.
- [66] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [67] A. M. Tillmann and M. E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014.
- [68] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond nyquist: Efficient sampling of sparse bandlimited signals. *arXiv preprint arXiv:0902.0026*, 2009.
- [69] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. Convex optimization in julia. In *Proceedings of the 1st First Workshop for High Performance Technical Computing in Dynamic Languages*, pages 18–28. IEEE Press, 2014.
- [70] M. B. Wakin. Manifold-based signal recovery and parameter estimation from compressive measurements. *arXiv preprint arXiv:1002.1247*, 2010.
- [71] Y. Wiaux, L. Jacques, G. Puy, A. M. Scaife, and P. Vandergheynst. Compressed sensing imaging techniques for radio interferometry. *Monthly Notices of the Royal Astronomical Society*, 395(3):1733–1742, 2009.
- [72] W. Yin and Y. Zhang. Extracting salient features from less data via ℓ_1 -minimization. *SIAG/OPT Views-and-News*, 19(1):11–19, 2008.

- [73] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing. *SIAM Journal on Imaging sciences*, 1(1):143–168, 2008.
- [74] Y. Zhang. Theory of compressive sensing via ℓ_1 -minimization: a non-rip analysis and extensions. *Journal of the Operations Research Society of China*, 1(1):79–105, 2013.

APÉNDICES

Apéndice 1

Código utilizado

En este apéndice se incluye una breve descripción del código utilizado para realizar las distintas pruebas. Dicho código está escrito en el lenguaje de programación Julia.

- *ejecutable_cluster.sh*: Se utiliza únicamente para ejecutar las pruebas en el cluster FIng. Se debe ejecutar desde la carpeta con el código Julia y mediante el comando: `qsub executable_cluster.sh`.
- *ejecutable_julia.jl*: Programa principal del código Julia. En este archivo se selecciona el tipo de prueba a realizar y el algoritmo con el cual se realizarán.
- *asignar_pruebas.jl*: Asigna las pruebas a realizar con el algoritmo seleccionado. Para esto llama repetidas veces a la función definida en *pruebas.jl*, con distintos valores en los parámetros, pero manteniendo el mismo algoritmo. Se asigna una ejecución de la función pruebas a cada procesador disponible. Es decir que, cada vez que un procesador se encuentra disponible, se le asignan N_p pruebas.
- *generar_problema.jl*: Genera problema Compressed Sensing de tamaño $m \times n$, y con solución tal que $\|x^*\|_0 \leq k$.
- *pruebas.jl*: Genera N_p problemas CS con los parámetros m , n y k recibidos. Para cada problema generado, aplica el algoritmo RW seleccionado y mide su porcentaje de recuperación (o la mejora respecto a la minimización ℓ_1 si se modela con ruido).
- *rw1_cwb.jl*: Algoritmo $RW\ell_1$ propuesto por Candès, Wakin y Boyd en [20].
- *rw1_subgrad.jl*: Algoritmo $RW\ell_1$ propuesto en este trabajo, implementado con subgradiente.
- *sin_rw.jl*: Resuelve problema de minimización ℓ_1 (sin RW) y con pesos unidad. Para esto ejecuta el código *wl1_reformulado.jl*, con $w = \vec{1}$.
- *w_lasso_fista.jl*: Algoritmo W-FISTA utilizado para resolver problema (W-LASSO).
- *rw_subgrad_lasso.jl*: Algoritmo RW-LASSO propuesto en este trabajo, implementado con subgradiente.

- *wl1_reformulado.jl*: Resuelve el problema Weighted ℓ_1 asociado a un vector de pesos $w \geq 0$. Para esto lo reformula como un LP o un SOCP, según si se modela sin o con ruido, respectivamente. Para resolver estos últimos problemas utiliza el solver *ECOS* de Julia.
- *prox_w.g.jl*: Implementa el operador proximal de la función ℓ_1 con pesos: $c\|Wx\|_1$, $c > 0$ y $W = \text{diag}(w)$, $w \geq 0$.
- *rip_2k.jl*: Halla mínima cantidad m de medidas que garantizan cumplir RIP $2k$ con $\delta_{2k} < \frac{4}{\sqrt{41}}$.