FACULTAD
DE
CIENCIAS

Recuperación de soluciones sparse usando métodos de optimización de puntos interiores

Sparse Solution Recovery using Interior Point Optimization Methods

Trabajo de fin de Grado
para acceder al

GRADO EN MATEMÁTICAS

Autora: Ana López Ríos

Directora: Cecilia Pola Méndez

Santander, 17 de octubre de 2016

Índice general

Resumen

En este trabajo vamos a considerar la recuperación de soluciones sparse para sistemas de ecuaciones lineales indeterminados. Estudiaremos la unicidad de solución del problema. Para su resolución numérica plantearemos varias formulaciones de optimización y analizaremos sus propiedades teóricas. Aquellos problemas que involucren a la norma $\|\cdot\|_1$ serán resueltos mediante métodos de optimización de puntos interiores. Además presentaremos algunos resultados numéricos obtenidos con dos códigos: linprog (Optimization Toolbox, MATLAB) y l1eq_pd (l_1-MAGIC). Se mostrará la influencia del escalamiento y de la aleatoriedad de la muestra en la solución recuperada. Algunos de los experimentos tratarán de la recuperación de una señal sonora.
Palabras clave: sistemas de ecuaciones lineales indeterminados, soluciones sparse, métodos de optimización de puntos interiores.

Abstract

In this work we are going to consider the sparse solution recovery for underdetermined systems of linear equations. We will study the uniqueness of the sparsest solution. For the numerical solution of the problem we will set out several optimization formulations and we will analyze their theoretical properties. Interior point optimization methods are used for solving those problems which involve the $\|\cdot\|_1$ norm. Moreover we will present some numerical results obtained with two codes: linprog (Optimization Toolbox, MATLAB) and l1eq_pd (l_1-MAGIC). We will also show the influence of the scaling and the random sampling on the recovered solution. Some of the experiments will be about signal recovery.
Key words: underdetermined systems of linear equations, sparse solutions, interior point optimization methods.

Capítulo 1

Introducción

En este trabajo nos ocuparemos de la recuperación de vectores sparse a partir de un pequeño número de observaciones. Concretamente trataremos el problema de la recuperación de un vector, $x_0 \in \mathbb{R}^n$, a partir de un vector $b \in \mathbb{R}^m$ con $m \ll n$ y tal que $b = Ax_0$ donde $A \in \mathbb{R}^{m \times n}$ tiene rango máximo ($rango(A) = m$) y también es conocida. Estamos interesados entonces en la resolución de sistemas de ecuaciones lineales indeterminados.
En ciencia y tecnología existen muchas situaciones que necesitan resolver este tipo de sistemas. Se sabe que estos sistemas tienen infinitas soluciones, sin embargo muchas de estas aplicaciones lo que buscan realmente son soluciones que tengan una propiedad adicional, ser sparse, es decir, soluciones con pocos elementos distintos de cero. La propiedad de ser sparse es muy útil en la práctica, por ejemplo, una imagen puede tener muchos megapixels, pero si la escribimos en una base adecuada, muchos de los coeficientes se convierten en magnitudes insignificantes que pueden considerarse iguales a cero (esto está en la base de los algoritmos de compresión como JPEG y JPEG2000), es decir, es sparse.
Al empezar a tratar este tema surgen algunas preguntas: ¿podemos encontrar una solución única? y en el caso de que la respuesta sea afirmativa, ¿bajo qué condiciones? Estas preguntas suscitaron mucho interés en el año 2004 (ver [4]) y surgieron para buscar sustento teórico a una serie de fenómenos que en principio parecían sorprendentes: había experiencias en sismología y en resonancia magnética donde se recuperaban imágenes a partir de mucha menos información de lo teóricamente estipulado. Este problema fue estudiado por Emmanuel Candes, Terence Tao, Justin Romberg y David Donoho, quienes determinaron, considerando las señales representadas por vectores de n componentes, que si se conoce que una señal es k sparse (en una determinada base tiene sólamente k coeficientes distintos de cero) y se tiene una muestra de la señal dada por $b = Ax$ para alguna matriz A, puede recuperarse la señal si ella es suficientemente sparse (ver [4]).
Sin embargo, una cosa es saber que existe solución y otra es poder calcularla numéricamente. De hecho, el problema de optimización que directamente se formula para calcular soluciones sparse optimales es un problema que, en general, es inviable de resolver en la práctica y requiere considerar formulaciones alternativas. Surge así la necesidad de estudiar en qué casos esas formulaciones nos permiten obtener la solución más sparse.
A continuación veamos algunas definiciones básicas para este trabajo.

Definición 1.0.1. *Un* **vector sparse o poco denso** *es un vector en el que el número de las coordenadas iguales a cero es relativamente grande en relación a su dimensión.*

Definición 1.0.2. *La* **densidad de un vector** x *es la cantidad de coordenadas distintas de cero que tiene dicho vector.*

Esta memoria está estructurada en ocho capítulos:

- En el siguiente capítulo analizaremos distintos problemas de optimización con la finalidad de, mediante su resolución numérica, calcular soluciones sparse. Aquellos que involucren

a la norma $\|\cdot\|_1$ serán transformados en problemas de programación lineal, abordando su resolución mediante la aplicación de métodos de puntos interiores.

- En el tercer capítulo trataremos la unicidad de soluciones sparse, es decir, estableceremos alguna condición bajo la cual podemos asegurar que la solución calculada es la más sparse de todas las posibles. Para ello introduciremos el concepto de número spark de una matriz.

- En el cuarto capítulo introduciremos los métodos de puntos interiores que vamos a utilizar para resolver en la práctica el problema planteado. En particular presentaremos el esquema general de los métodos primales-duales, y consideraremos las características particulares de los dos que vamos a usar en la experimentación numérica (implementados en el software l_1-MAGIC [2] y en la función linprog de MATLAB). También definiremos el concepto de camino central, demostraremos su existencia y los entornos del mismo más utilizados.

- En el quinto capítulo nos centraremos en la experimentación numérica con el objetivo de recuperar soluciones sparse con distintas formulaciones y dos programas, l1eq_pd incluido en l_1-MAGIC y linprog citado anteriormente. En primer lugar trabajaremos con problemas generados aleatoriamente para pasar a continuación a tratar de recuperar una señal. Nos planteamos distintas formas de tomar las observaciones y el número de estas, mostrándose la utilidad de tomarlas aleatoriamente.

- Terminamos la memoria con tres capítulos, el primero dedicado a la exposición de unas conclusiones sobre el trabajo realizado, el segundo recopila algunos resultados de optimización en relación a las condiciones de optimalidad de primer orden y la existencia de soluciones óptimas y, en el último adjuntamos una tabla que expresa las notaciones y la terminología utilizada en este trabajo para evitar cualquier confusión.

Capítulo 2

Problemas de optimización para calcular soluciones sparse

Como ya hemos introducido en el capítulo anterior, nuestro problema va a tratar de la recuperación de un vector $x_0 \in \mathbb{R}^n$, a partir de un pequeño número de observaciones $b = Ax_0 \in \mathbb{R}^m$, donde $m \ll n$ y $A \in \mathbb{R}^{m\times n}$ es una matriz de rango máximo m. Por lo tanto se trata de resolver un sistema de ecuaciones lineales indeterminado, $Ax = b$, el cual tiene infinitas soluciones. Sin embargo nosotros vamos a buscar la solución más sparse, es decir, la que tiene mayor número de coordenadas iguales a cero. Nos preguntamos entonces si este problema tiene una única solución y, si esto ocurre, bajo qué condiciones podemos encontrarla. Para tratar de resolver el problema planteado vamos a analizar las tres formulaciones siguientes en las que tomaremos distintas funciones objetivo.

$$(P_0)\begin{cases} \text{mín} & J_0(x) = \sum_{i=1}^{n} |x_i|^0 \\ x \in \mathbb{R}^n & \\ Ax = b & \end{cases}, \tag{2.1}$$

donde usaremos la convención de que $0^0 = 0$. Es decir, evaluar la función objetivo en un vector, x, consiste simplemente en contabilizar el número de coordenadas de x que son distintas de cero.

$$(P_1)\begin{cases} \text{mín} & J_1(x) = \|x\|_1 \\ x \in \mathbb{R}^n & \\ Ax = b & \end{cases}. \tag{2.2}$$

$$(P_2)\begin{cases} \text{mín} & J_2(x) = \frac{1}{2}\|x\|_2^2 \\ x \in \mathbb{R}^n & \\ Ax = b & \end{cases}. \tag{2.3}$$

A continuación analizamos las propiedades de los tres problemas anteriores. Empecemos por (P_0).

Proposición 2.0.1. *La función J_0 no es continua.*

Demostración. Para el caso n=1 la función toma los valores:

$$J_0(x) = \begin{cases} 0 & si \quad x = 0 \\ 1 & si \quad x \neq 0 \end{cases}. \tag{2.4}$$

□

Para el caso n=2 la función viene dada por:

$$J_0(x) = \begin{cases} 0 & si \quad x = (0,0)^T, \\ 1 & si \quad x_1 x_2 = 0 \quad \text{y} \quad |x_1| + |x_2| > 0, \\ 2 & si \quad x_i \neq 0 \ \ \forall i \in \{1,2\}. \end{cases}$$

Notemos que para el problema (P_0) hay soluciones locales que no son globales. Por ejemplo, para el problema asociado a la ecuación $x_1 + x_2 = 1$, (0,5, 0,5) es solución local pero no es solución global.
A continuación veremos que (P_1) no sufre esos inconvenientes.

Proposición 2.0.2. *Existe solución para el problema (P_1) y además toda solución local es global.*

Demostración. Veamos en primer lugar la existencia de solución. Para ello basta notar que:

1. El conjunto $K = \{x \in \mathbb{R}^n : Ax = b\}$ es un conjunto cerrado.

2. $J_1(x) = \|x\|_1$ es una función coerciva y continua.

Utilizando la proposición 7.2.1 del anexo ya tenemos asegurada la existencia de solución.
Para ver que toda solución local es global basta aplicar la proposición 7.2.2 del anexo, notando que la función $J_1(x) = \|x\|_1$ es una función convexa y K es un conjunto convexo. □

Para evitar la no diferenciabilidad de la función objetivo de (P_1), vamos a considerar un problema asociado a (P_1) pero que nos elimine el valor absoluto (ver por ejemplo [5]). Para ello vamos a tomar unas nuevas variables $u \in \mathbb{R}^n$ asociadas a $x \in \mathbb{R}^n$ tales que $u_i = |x_i|$. De este modo nos queda la formulación siguiente:

$$(\tilde{P}_1) \begin{cases} \text{mín} \qquad \tilde{J}(x,u) = \sum_{i=1}^{n} u_i \\ (x,u) \in \mathbb{R}^n \times \mathbb{R}^n \\ Ax = b \\ x_i - u_i \leqslant 0, \quad i = 1, \ldots, n \\ -x_i - u_i \leqslant 0, \quad i = 1, \ldots, n \end{cases} . \tag{2.5}$$

Proposición 2.0.3. *Se verifican las siguientes propiedades:*

a. Existe solución para $(\tilde{P}_1)$.

b. Si $(\tilde{x}, \tilde{u})$ es solución de $(\tilde{P}_1)$, entonces $\tilde{x}$ es solución global de (P_1).

Demostración. a. En primer lugar el conjunto de puntos admisibles [1], $\mathcal{A}(\tilde{P}_1)$, es un conjunto cerrado y la función objetivo es continua. Además si probamos que $\tilde{J}$ es coerciva en $\mathcal{A}(\tilde{P}_1)$, aplicando la proposición 7.2.1 del anexo podemos afirmar que existe al menos una solución para $(\tilde{P}_1)$.
Veamos que efectivamente la función $\tilde{J}$ es coerciva en $\mathcal{A}$ $(\tilde{P}_1)$: dado (x,u) admisible para $(\tilde{P}_1)$, podemos asegurar que $u_i \geqslant |x_i| \geqslant 0$ para $i = 1, \ldots, n$ y, por tanto

$$\tilde{J}(x,u) = \sum_{i=1}^{n} u_i = \frac{1}{2}\sum_{i=1}^{n} u_i + \frac{1}{2}\sum_{i=1}^{n} u_i \geqslant \frac{1}{2}\|u\|_1 + \frac{1}{2}\sum_{i=1}^{n} |x_i| = \frac{1}{2}[\|u\|_1 + \|x\|_1] = \frac{1}{2}\|(u,x)\|_1.$$

b. Dado $x \in \mathcal{A}$ (P_1), entonces el par $(x,|x|)$ es admisible para $(\tilde{P}_1)$ y, como $(\tilde{x}, \tilde{u})$ es solución global de $(\tilde{P}_1)$ se tiene: $\tilde{J}(x,|x|) \geqslant \tilde{J}(\tilde{x}, \tilde{u})$, o lo que es lo mismo: $\sum_{i=1}^{n} |x_i| \geqslant \sum_{i=1}^{n} \tilde{u}_i$. De donde, usando que $\tilde{u}_i \geqslant |\tilde{x}_i|$, obtenemos: $J_1(x) \geqslant J_1(\tilde{x})$. De donde, como además $\tilde{x} \in \mathcal{A}$ (P_1), $\tilde{x}$ es solución global de (P_1).

□

[1] puntos que verifican las restricciones del problema

Notemos que $(\tilde{P}_1)$ es un problema de programación lineal y por tanto más fácil de resolver que (P_1). Notemos también que tiene el doble de variables que (P_1). Se pueden considerar otras formulaciones de programación lineal para resolver (P_1). Por ejemplo (ver [7]), considerando que $|x_i| = u_i + v_i$ con $u_i = \text{máx}\{0, x_i\}$ y $v_i = \text{máx}\{0, -x_i\}$ llegamos a obtener el siguiente problema:

$$(\tilde{\tilde{P}}_1) \begin{cases} \text{mín} & \tilde{\tilde{J}}_1(u,v) = \sum_{i=1}^{n}(u_i + v_i) \\ (u,v) \in \mathbb{R}^n \times \mathbb{R}^n & \\ A(u-v) = b & \\ u_i \geqslant 0, \quad i = 1,\ldots,n & \\ v_i \geqslant 0, \quad i = 1,\ldots,n & \end{cases}.$$

Proposición 2.0.4. *Se tiene que:*

a. Existe solución global para el problema $(\tilde{\tilde{P}}_1)$.

b. Si $(\tilde{u}, \tilde{v})$ *es solución de* $(\tilde{\tilde{P}}_1)$*, entonces* $\tilde{u} - \tilde{v}$ *es solución global de* (P_1).

Demostración. a. Basta aplicar la proposición 7.2.4 del anexo.

b. Dado $x \in \mathcal{A}$ (P_1), entonces $(\text{máx}\{0,x\}, \text{máx}\{0,-x\})$ es admisible para el problema $(\tilde{\tilde{P}}_1)$ y, como sabemos que $(\tilde{u}, \tilde{v})$ es solución global de $(\tilde{\tilde{P}}_1)$, se tiene que:

$$\tilde{\tilde{J}}_1(\text{máx}\{0,x\}, \text{máx}\{0,-x\}) \geqslant \tilde{\tilde{J}}_1(\tilde{u}, \tilde{v}).$$

De donde usando que $\tilde{u} \geqslant 0$, $\tilde{v} \geqslant 0$ y la desigualdad triangular, se tiene

$$\sum_{i=1}^{n} |x_i| \geqslant \sum_{i=1}^{n}(\tilde{u}_i + \tilde{v}_i) = \sum_{i=1}^{n} |\tilde{u}_i| + |\tilde{v}_i| \geqslant \sum_{i=1}^{n} |\tilde{u}_i - \tilde{v}_i| = J_1(\tilde{u} - \tilde{v}).$$

Por lo tanto ha quedado demostrado que:

$$J_1(x) \geqslant J_1(\tilde{u} - \tilde{v})$$

y por tanto $\tilde{u} - \tilde{v}$ es solución global del problema (P_1).

□

De este modo hemos transformado el problema (P_1) en un problema de programación lineal que tiene forma estándar (solo tiene restricciones generales de igualdad y todas las variables son mayores o iguales que cero).
En los dos siguientes resultados estudiamos la unicidad de solución para los problemas (P_1) y (P_2).

Proposición 2.0.5. *En general el problema* (P_1) *no tiene unicidad de solución.*

Demostración. Veamos un ejemplo para n=2:

$$\begin{cases} \text{mín} & |x_1| + |x_2| \\ (x_1, x_2)^T \in \mathbb{R}^2 & \\ -x_1 + x_2 = -2 & \end{cases}$$

Si tomamos la restricción $x_2 = x_1 - 2$ entonces el problema anterior pasa a ser:

$$\begin{cases} \text{mín} & |x_1| + |x_1 - 2| \\ x_1 \in \mathbb{R} & \end{cases}$$

que tiene infinitas soluciones: el intervalo $[0, 2]$. □

Proposición 2.0.6. *El problema (P_2) tiene una única solución.*

Para demostrarlo necesitaremos el siguiente lema:

Lema 2.0.7. *La matriz AA^T es inversible.*

Demostración. Para ello vamos a ver que la matriz AA^T es definida positiva. Probemos que: $d^T(AA^T)d > 0, \quad \forall d \in \mathbb{R}^m \setminus 0$. Usando que $d^T AA^T d = \|A^T d\|_2^2 \geqslant 0$, basta ver que $\|A^T d\|_2^2 \neq 0$. Esto se deduce de las siguientes implicaciones: $\|A^T d\|_2^2 = 0 \Rightarrow A^T d = 0 \Rightarrow d = 0$ por ser el $rango(A^T) = m$.

□

Demostración. En primer lugar el conjunto de restricciones $K = \{x \in \mathbb{R}^n : Ax = b\}$ es un conjunto convexo. Además nuestra función $J_2(x) = \frac{1}{2}\|x\|_2^2$ es estrictamente convexa en K y por tanto, por las condiciones suficientes de optimalidad de primer orden (ver teorema 7.1.4 del anexo), tenemos que si $\bar{x}$ es un punto de Kuhn-Tucker del problema (P_2) (ver definición 7.1.3 en el anexo), entonces $\bar{x}$ es solución global del problema.
Veamos ahora qué tiene que cumplir $\bar{x}$ para ser punto de Kuhn-Tucker del problema (P_2).
La condición (7.3) en este caso toma la forma: $\bar{x} + A^T\lambda = 0$. Esto unido a (7.5) nos queda que $\bar{x}$ tiene que cumplir el siguiente sistema:

$$\begin{cases} \bar{x} + A^T\lambda = 0 \\ A\bar{x} = b \end{cases} .$$

Multiplicamos a la primera igualdad por A y nos queda: $A\bar{x} + AA^T\lambda = 0$ y usando ahora la segunda igualdad deducimos: $b + AA^T\lambda = 0$. De donde utilizando el lema 2.0.7, obtenemos: $\lambda = -(AA^T)^{-1}b$. Volvemos a la primera igualdad del sistema y sustituimos λ obteniendo: $\bar{x} + A^T(-(AA^T)^{-1}b) = 0$. De donde concluimos que el único punto de Kuhn-Tucker es: $\bar{x} = A^T(AA^T)^{-1}b$.
La unicidad de la solución está asegurada por el teorema 7.2.3. □

En conclusión hemos considerado tres formulaciones. La resolución de (P_0) implica resolver un problema combinatorio, cuya complejidad computacional hace inviable su resolución práctica en general. El problema (P_1) puede considerarse una relajación convexa de (P_0) que puede ser transformada en un problema de programación lineal, siendo por tanto más asequible su resolución. Además, como veremos en el siguiente capítulo, cuando la solución de (P_0) es suficientemente sparse, es la misma que la solución del problema (P_1) (ver [10]).
Para resolver el problema (P_1) utilizaremos métodos de puntos interiores que introduciremos en el cuarto capítulo. Nosotros emplearemos la función l1eq_pd del software l_1- MAGIC (que está basado en la formulación $(\tilde{P}_1)$) y la función linprog de MATLAB con las dos formulaciones de programación lineal planteadas, $(\tilde{P}_1)$ y $(\tilde{\tilde{P}}_1)$. Finalmente, para abordar el problema (P_2) emplearemos la función pinv de MATLAB.
En este capítulo hemos analizado tres formulaciones distintas para tratar de resolver nuestro problema, pero sería natural preguntarse por qué no utilizamos otras formulaciones:

$$(P_p) \begin{cases} \text{mín} \quad \|x\|_p \\ x \in \mathbb{R}^n \\ \text{sujeto a } Ax = b \end{cases} , \tag{2.6}$$

para valores de p tales que $0 < p < 1$. Desafortunadamente en esos casos nos encontramos con problemas de optimización no convexos, los cuales son muy difíciles de resolver en general. En la siguiente figura se representa el comportamiento de la función $|x|^p$ para varios valores de p. Para $p = 2$ y $p = 1$ se obtienen funciones convexas. No ocurre lo mismo para $p < 1$.

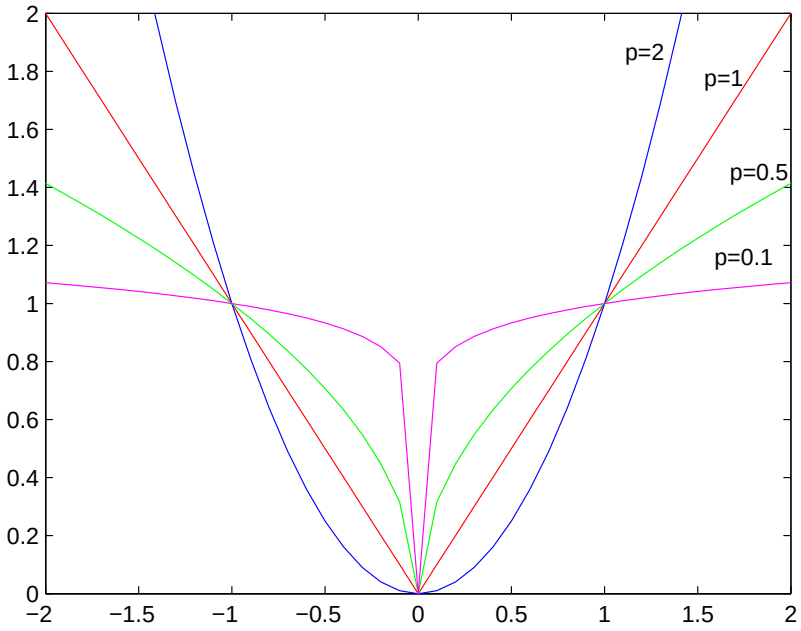

Figura 2.1: Representación de la función $|x|^p$ para $x \in [-2, 2]$ y para distintos valores de p.

Capítulo 3

Unicidad de las soluciones sparse

En este capítulo estableceremos un resultado teórico que nos garantice la unicidad de solución para el problema (P_0), esto es, soluciones del sistema planteado que tengan el menor número de coordenadas distintas de cero. En primer lugar vamos a introducir un concepto nuevo: el número spark de una matriz (ver [3]).

Definición 3.0.1. El número spark de una matriz, *A, es el menor número de columnas de A que son linealmente dependientes y se representa por spark(A).*

Recordemos que el rango de una matriz está definido como el máximo número de columnas de la matriz que son linealmente independientes. Estas dos definiciones pueden tener un cierto parecido, pero es mayor la dificultad de calcular el número spark, dado que hay que hacer una búsqueda combinatoria entre todos los posibles subconjuntos que forman las columnas de A.
Notemos que si $A \in \mathbb{R}^{m\times n}$ es una matriz de rango máximo m, entonces spark(A) es un número natural del intervalo [1, m+1]. Además si $x \in \mathbb{R}^n$ es tal que $Ax = 0$, entonces satisface que $\|x\|_0 \geqslant$ spark(A).
A continuación se establece un criterio de unicidad de solución para el problema (P_0), gracias al concepto de número spark (ver [3]).

Teorema 3.0.2. *Si un sistema de ecuaciones lineales, $Ax = b$, tiene una solución, $\bar{x}$, que cumple que $\|\bar{x}\|_0 < spark(A)/2$, entonces esa solución es necesariamente la más sparse posible.*

Demostración. Suponemos que existe otra solución, y. Se satisface que $Ay = b$ y, usando que $A\bar{x} = b$, se deduce que $A(\bar{x} - y) = 0$. Ahora tomando la definición de número spark sabemos que:

$$\|\bar{x} - y\|_0 \geqslant spark(A).$$

Además, $\|\bar{x}\|_0 + \|y\|_0 \geqslant \|\bar{x} - y\|_0$, esto es, el número de coordenadas distintas de cero del vector $\bar{x} - y$ es menor o igual que la suma de las coordenadas distintas de cero en $\bar{x}$ y en y. Así que considerando las dos desigualdades anteriores y que por hipótesis $\|\bar{x}\|_0 < \frac{spark(A)}{2}$, necesariamente se tiene que cumplir que $\|y\|_0 > \frac{spark(A)}{2}$ y por tanto y no será la solución más sparse. □

Como en la práctica, en general, no es sencillo calcular el número spark, tiene interés determinar otras formas más simples de garantizar la unicidad de solución. Al menos hasta donde nosotros sabemos (ver [3]), por el momento las alternativas dan lugar a resultados mucho menos potentes que el presentado.

Capítulo 4

Métodos de puntos interiores

Para abordar los problemas de programación lineal planteados en el segundo capítulo utilizaremos los métodos de puntos interiores que explicaremos a continuación.
En los años 80 se descubrió que problemas de programación lineal de talla grande podían ser resueltos eficientemente usando técnicas de programación no lineal. Una característica propia de estos métodos es que todas las iteraciones deben satisfacer las restricciones de desigualdad estrictamente, por eso fueron nombrados **métodos de puntos interiores**. Estos métodos surgieron durante la búsqueda de algoritmos con mejores propiedades teóricas que el método del simplex, que para algunos problemas tenía un comportamiento exponencial.
En los años 90 surgieron los métodos primales-duales que se convirtieron en los métodos de puntos interiores más eficientes en la práctica, resultando ser grandes competidores del método del simplex cuando se trataba de problemas de talla grande.
Los métodos de puntos interiores tienen características comunes que los diferencian del método del simplex. Cada iteración de los métodos de puntos interiores tiene un gran coste pero puede hacer un gran progreso hacia la solución, mientras que el método del simplex normalmente requiere de un número mayor de iteraciones más baratas de realizar. Geométricamente el método simplex recorre la frontera del conjunto admisible, probando una serie de vértices hasta que encuentra el óptimo, mientras que los métodos de puntos interiores no buscan soluciones en la frontera, sino que las alcanzan desde el interior.
Los métodos de puntos interiores tienen asociada una extensa bibliografía de la cual destacamos dos referencias para introducirse en el tema: un libro de optimización general ([1]) y un libro especializado en el tema ([6]).

4.1. Métodos Primales-Duales

Consideramos el problema de programación lineal en la forma estándar:

$$\begin{cases} \text{mín} & c^T x \\ x \in \mathbb{R}^n & \\ \tilde{A}x = \tilde{b} & \\ x \geqslant 0 & \end{cases}, \tag{4.1}$$

donde $c \in \mathbb{R}^n, \tilde{b} \in \mathbb{R}^m$ y $\tilde{A} \in \mathbb{R}^{m \times n}$ de rango máximo: $rango(\tilde{A}) = m$.
El problema dual asociado es:

$$\begin{cases} \text{máx} & \tilde{b}^T \lambda \\ \lambda \in \mathbb{R}^m, s \in \mathbb{R}^n & \\ \text{sujeto a} & \tilde{A}^T \lambda + s = c \\ s \geqslant 0 & \end{cases}. \tag{4.2}$$

Las condiciones de Kuhn-Tucker del problema (4.1) dan lugar al siguiente sistema de ecuaciones no lineales e inecuaciones:

$$\tilde{A}^T\lambda + s = c, \tag{4.3}$$

$$\tilde{A}x = \tilde{b}, \tag{4.4}$$

$$x_i s_i = 0, \quad 1 \leqslant i \leqslant n, \tag{4.5}$$

$$(x, s) \geqslant 0. \tag{4.6}$$

Como vemos las condiciones (4.3), (4.4) y (4.6) hacen referencia a la admisibilidad de los dos problemas, el primal y el dual. En programación lineal las condiciones de Kuhn-Tucker son condiciones necesarias y suficientes para las soluciones de los problemas. Por ello, los métodos numéricos en este ámbito utilizan estas condiciones como criterio de optimalidad.
Para introducir los métodos de puntos interiores veamos los puntos de Kuhn-Tucker como ceros de la siguiente función:

$$F(x, \lambda, s) = \begin{bmatrix} \tilde{A}^T\lambda + s - c \\ \tilde{A}x - \tilde{b} \\ XSe \end{bmatrix}, \tag{4.7}$$

que además tienen que verificar:

$$(x, s) \geqslant 0, \tag{4.8}$$

donde X y S son matrices diagonales tales que $X_{ii} = x_i$ y $S_{ii} = s_i$ para $i = 1, 2, \ldots, n$ y $e = (1, 1, \ldots, 1)^T$.
Como todos los algoritmos iterativos en optimización, estos métodos tienen dos ingredientes básicos: un procedimiento para determinar la dirección y otro para el paso o desplazamiento. Para determinar la dirección usamos el método de Newton para sistemas de ecuaciones no lineales, que, dado un iterante (x, λ, s), propone el cálculo de una dirección, $(\Delta x, \Delta\lambda, \Delta s)$, resolviendo el siguiente sistema de ecuaciones lineales:

$$J(x, \lambda, s) \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -F(x, \lambda, s), \tag{4.9}$$

donde J es el jacobiano de F o explícitamente:

$$\begin{pmatrix} 0 & \tilde{A}^T & I \\ \tilde{A} & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\tilde{A}^T\lambda - s + c \\ -\tilde{A}x + \tilde{b} \\ -XSe \end{pmatrix}. \tag{4.10}$$

Veamos que la dirección está unívocamente determinada en el siguiente resultado.

Lema 4.1.1. *Si $x > 0$ y $s > 0$, entonces la matriz*

$$\begin{pmatrix} 0 & \tilde{A}^T & I \\ \tilde{A} & 0 & 0 \\ S & 0 & X \end{pmatrix} \tag{4.11}$$

es inversible.

Demostración. Para ello vamos a ver que el siguiente sistema de ecuaciones

$$\tilde{A}^T d_\lambda + I d_s = 0 \tag{4.12}$$

$$\tilde{A} d_x = 0 \tag{4.13}$$

$$S d_x + X d_s = 0 \tag{4.14}$$

implica que $d_x = 0, d_\lambda = 0$ y $d_s = 0$.

De la ecuación (4.14) se deduce que $d_s = -(X)^{-1}Sd_x$, que lo sustituimos en la primera ecuación quedando del siguiente modo: $\tilde{A}^T d_\lambda - (X)^{-1}Sd_x = 0$ y equivalentemente $(X^{-1}S)^{-1}\tilde{A}^T d_\lambda = d_x$, multiplicando ahora por $\tilde{A}$ al lado izquierdo obtenemos: $\tilde{A}S^{-1}X\tilde{A}^T d_\lambda = \tilde{A}d_x$ y, usando (4.13), tenemos que: $\tilde{A}S^{-1}X\tilde{A}^T d_\lambda = 0$ lo que podemos descomponer de la siguiente forma,

$$(\tilde{A}X^{\frac{1}{2}}S^{-\frac{1}{2}}S^{-\frac{1}{2}}X^{\frac{1}{2}}\tilde{A}^T)d_\lambda = 0,$$

dado que las matrices S y X son diagonales y por tanto se pueden conmutar. De este modo tenemos: $(\tilde{A}X^{\frac{1}{2}}S^{-\frac{1}{2}})(\tilde{A}X^{\frac{1}{2}}S^{-\frac{1}{2}})^T d_\lambda = 0$ y por ser $(\tilde{A}X^{\frac{1}{2}}S^{-\frac{1}{2}})^T$ de rango máximo por columnas se deduce que $d_\lambda = 0$. De ahí, usando (4.12), $d_s = 0$ y con (4.14) se deduce que $d_x = 0$. □

Para determinar el desplazamiento a lo largo de la dirección calculada, α, hay que tener en cuenta que tiene que cumplirse (4.8), para lo que se hace una búsqueda de línea tomando los iterantes del siguiente modo:

$$(x, \lambda, s) + \alpha(\Delta x, \Delta\lambda, \Delta s) \text{ para un desplazamiento } \alpha \in (0, 1].$$

La mayoría de los métodos primales-duales usan una modificación de (4.10) (el esquema de Newton) para tratar de evitar desplazamientos demasiado pequeños. Esta variación consiste en usar las ecuaciones $x_i s_i = \sigma\mu$ para $i = 1, \ldots, n$, donde μ es la **medida de dualidad** definida por:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n} \tag{4.15}$$

y evaluada en el punto de partida y $\sigma \in [0, 1]$ es el factor de reducción en la medida de dualidad que queremos conseguir. De este modo el sistema de ecuaciones (4.10) queda definido del siguiente modo:

$$\begin{pmatrix} 0 & \tilde{A}^T & I \\ \tilde{A} & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\tilde{A}^T\lambda - s + c \\ -\tilde{A}x + \tilde{b} \\ -XSe + \sigma\mu e \end{pmatrix} \tag{4.16}$$

donde σ se denomina **parámetro de centrado**.

De este modo el **esquema de los métodos primales-duales** es el siguiente:

Dado (x^0, λ^0, s^0) con $(x^0, s^0) > 0$,
para $k = 0, 1, 2, \ldots$
elegimos $\sigma_k \in [0, 1]$ y calculamos una dirección resolviendo :

$$\begin{pmatrix} 0 & \tilde{A}^T & I \\ \tilde{A} & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} -\tilde{A}^T\lambda^k - s^k + c \\ -\tilde{A}x^k + \tilde{b} \\ -X^k S^k e + \sigma_k\mu_k e \end{pmatrix} \tag{4.17}$$

donde $\mu_k = (x^k)^T s^k/n$.
Entonces $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k(\Delta x^k, \Delta\lambda^k, \Delta s^k)$,
eligiendo un desplazamiento α_k tal que $(x^{k+1}, s^{k+1}) > 0$,
end.

Veamos por qué es importante que las variables x^k y s^k sean estrictamente positivas para cada iteración, en vez de ser únicamente no negativas. Dadas las ecuaciones $xs = 0$, las correspondientes ecuaciones de Newton en un punto dado, (x, s), son $x\Delta s + s\Delta x = -xs$. Si una variable s_i, es nula, entonces la ecuación de Newton quedaría como: $x_i\Delta s_i = 0$, lo cual nos lleva a $\Delta s_i = 0$. Como consecuencia la variable s_i permanecerá siendo nula a partir de que se haga cero por primera vez. Esto es un inconveniente dado que un algoritmo nunca será capaz de recuperarse una vez se haga cero una de las variables.

La elección del parámetro de centrado σ_k y de la longitud del paso α_k dan lugar a una amplia variedad de métodos con distintas propiedades. Nosotros utilizaremos dos esquemas.
El primero de ellos es el que usa el software l_1- MAGIC (ver [5]), en el que el desplazamiento debe proporcionar un decrecimiento suficiente del residuo:

$$\|F(x+\alpha\Delta x, \lambda+\alpha\Delta\lambda, s+\alpha\Delta s)\|_2 \leqslant (1-0{,}01\alpha)\|F(x,\lambda,s)\|_2.$$

Este algoritmo usa como test de parada $x^T s < \epsilon$, donde ϵ es una tolerancia dada.
El segundo esquema que vamos a utilizar en este trabajo es el que utiliza la función linprog de MATLAB. Este está basado en LIPSOL (ver [9]) que es una variante del algoritmo predictor-corrector de Mehrotra, en el que en cada iteración se calculan dos direcciones: una predictora, $\Delta^p v^k$, y otra correctora, $\Delta^c v^k$, resolviendo dos sistemas de ecuaciones lineales con la misma matriz como vemos en el siguiente esquema.

Algoritmo predictor-corrector

Sea $v = (x, \lambda, s)$
Eligimos un punto inicial v^0 tal que $(x^0, s^0) > 0$.
$k = 0$
Repetir hasta que $\|F(v^k)\|$ sea "pequeño", siendo F la función definida en (4.7).

1. Resolver $J(v^k)\Delta^p v^k = -F(v^k)$ y calcular $\mu^k = (x^k)^T s^k/n$.
2. Resolver $J(v^k)\Delta^c v^k = -F(v^k + \Delta^p v^k) + \begin{pmatrix} 0 \\ 0 \\ \mu^k \end{pmatrix}$.
3. Elegir $\alpha^k > 0$ y calcular $v^{k+1} = v^k + \alpha^k(\Delta^p v^k + \Delta^c v^k)$ tal que $x^{k+1} > 0$ y $s^{k+1} > 0$.

$k = k + 1$
end

El código linprog usa el siguiente test de parada:

$$\frac{\|\tilde{A}x^k - \tilde{b}\|}{\text{máx}(1, \|\tilde{b}\|)} + \frac{\|\tilde{A}^T\lambda^k + s^k - c\|}{\text{máx}(1, \|c\|)} + \frac{|c^T x^k - \tilde{b}^T\lambda^k|}{\text{máx}(1, |c^T x^k|, |\tilde{b}^T\lambda^k|)} \leqslant \bar{\epsilon},$$

donde $\bar{\epsilon}$ es una tolerancia dada. Esto es la suma total de los errores relativos del sistema de ecuaciones dado por las condiciones de Kuhn-Tucker. El último sumando está relacionado con la condición (4.5), ya que si x y (λ, s) son soluciones de los problemas primal-dual respectivamente, entonces $c^T x - \tilde{b}^T\lambda = s^T x$.

EL CAMINO CENTRAL

En esta sección introduciremos un concepto clave en el diseño de los métodos de puntos interiores primales-duales. Vamos a comenzar definiendo dos conjuntos:

Definición 4.1.2. *El* **conjunto primal-dual admisible** $\mathcal{F}$ *viene definido por:*

$$\mathcal{F} = \{(x,\lambda,s) | \tilde{A}x = \tilde{b}, \tilde{A}^T\lambda + s = c, (x,s) \geqslant 0\},$$

y el **conjunto primal-dual estrictamente admisible** *viene dado por:*

$$\mathcal{F}^o = \{(x,\lambda,s) | \tilde{A}x = \tilde{b}, \tilde{A}^T\lambda + s = c, (x,s) > 0\}.$$

El **camino central** $\mathcal{C}$ es un arco de puntos $(x_\tau, \lambda_\tau, s_\tau) \in \mathcal{F}^o$ parametrizado por un escalar positivo τ. Cada punto del camino $\mathcal{C}$ satisface las siguientes condiciones para algún $\tau > 0$:

$$\tilde{A}^T\lambda + s = c, \tag{4.18}$$

$$\tilde{A}x = \tilde{b}, \tag{4.19}$$

$$XSe = \tau e, \tag{4.20}$$

$$(x, s) > 0. \tag{4.21}$$

En los resultados teóricos que vamos a presentar se utilizará la hipótesis $\mathcal{F}^o \neq \emptyset$. Es por ello que hacemos notar en este punto que existen problemas de programación lineal que no verifican dicha condición. Veamos por ejemplo el siguiente problema:

$$\begin{cases} \text{mín} & x_1 \\ x \in \mathbb{R}^3 & \\ \text{tal que} & x_1 + x_3 = 0 \\ x \geqslant 0 & \end{cases},$$

cuyo problema dual sería

$$\begin{cases} \text{máx} & 0 \\ \lambda \in \mathbb{R}, s \in \mathbb{R}^3 & \\ \text{tal que} & \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \lambda + \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ s \geqslant 0 & \end{cases}.$$

El conjunto primal-dual estrictamente admisible asociado a este problema es el vacío porque toda solución satisface: $x_1 = x_3 = s_2 = 0$ por lo que ninguna cumplirá la desigualdad $(x, s) > 0$. Sin embargo el conjunto primal-dual admisible es el siguiente:

$$\mathcal{F} = \{(0, x_2, 0), \lambda, (1 - \lambda, 0, -\lambda) : \ x_2 \geqslant 0 \ \text{ y } \lambda \leqslant 0\}.$$

A continuación probaremos que el camino central está bien definido (ver [6]).

Teorema 4.1.3. *Sea $\mathcal{F}^o \neq \emptyset$ y $Z \in \mathbb{R}^{m\times(n-m)}$ de rango máximo tal que $\tilde{A}^T Z = 0$. Para cada $\tau > 0$ el problema*

$$(P_\tau) \begin{cases} \text{mín} & f_\tau(x, s) = \dfrac{1}{\tau} x^T s - \displaystyle\sum_{i=1}^{n} log(x_i s_i) \\ (x, s) \in \mathbb{R}^n \times \mathbb{R}^n & \\ \tilde{A}x = \tilde{b} & \\ Z^T s = Z^T c & \\ (x, s) > 0 & \end{cases} \tag{4.22}$$

tiene una única solución $(\bar{x}_\tau, \bar{s}_\tau)$. Además existe un único $\bar{\lambda}_\tau$ tal que $(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau)$ verifica las ecuaciones del camino central.

Para demostrar el teorema anterior probaremos un lema y dos proposiciones.

Lema 4.1.4. *Suponiendo que $\mathcal{F}^o \neq \emptyset$ y dado $\beta \geqslant 0$, entonces*

$$K_\beta = \{(x, s) \in \mathbb{R}^n \times \mathbb{R}^n \text{ tal que } (x, \lambda, s) \in \mathcal{F} \text{ y } \ x^T s \leqslant \beta\}$$

está acotado.

Demostración. Sea $(x,s) \in K_\beta$ cualquiera. Por pertenecer a K_β sabemos que se cumple que:

$$\tilde{A}x = \tilde{b} \tag{4.23}$$

$$\tilde{A}^T\lambda + s = c \tag{4.24}$$

$$(x,s) \geqslant 0 \tag{4.25}$$

y además $x^T s \leqslant \beta$. Para obtener el resultado probaremos que existe una constante M tal que $|x_i| \leqslant M$ y $|s_i| \leqslant M$ para $i = 1, \ldots, n$.
Como $\mathcal{F}^o \neq \emptyset$, podemos considerar $(\bar{x}, \bar{\lambda}, \bar{s}) \in \mathcal{F}^o$. Entonces se cumple que:

$$\tilde{A}\bar{x} = \tilde{b}, \tag{4.26}$$

$$\tilde{A}^T\bar{\lambda} + \bar{s} = c, \tag{4.27}$$

$$(\bar{x}, \bar{s}) > 0. \tag{4.28}$$

De (4.26) y (4.23) se deduce que:

$$\tilde{A}(\bar{x} - x) = 0. \tag{4.29}$$

De (4.27) y (4.24) se deduce que: $\tilde{A}^T(\bar{\lambda} - \lambda) + (\bar{s} - s) = 0$ y por tanto $(\bar{s} - s) = -\tilde{A}^T(\bar{\lambda} - \lambda)$. De donde $(\bar{x} - x)^T(\bar{s} - s) = (\bar{x} - x)^T(-\tilde{A}^T(\bar{\lambda} - \lambda)) = 0$ por (4.29). Tenemos por tanto que

$$\bar{x}^T\bar{s} - x^T\bar{s} - \bar{x}^T s + x^T s = 0$$

y despejando obtenemos que: $x^T\bar{s} + \bar{x}^T s = \bar{x}^T\bar{s} + x^T s$. Por hipótesis sabemos que $x^T s \leqslant \beta$ y por tanto podemos afirmar que: $x^T\bar{s} + \bar{x}^T s \leqslant \bar{x}^T\bar{s} + \beta$. Como, por otro lado, tenemos que $x^T\bar{s} + \bar{x}^T s \geqslant x^T\bar{s} \geqslant x_i\bar{s}_i$ para $i = 1, \ldots, n$, obtenemos que:

$$x_i \leqslant \frac{\bar{x}^T\bar{s} + \beta}{\bar{s}_i}.$$

Análogamente se llega a $s_i \leqslant \frac{\bar{x}^T\bar{s}+\beta}{\bar{x}_i}$.

Tomando $M = \max_{i=1}^{n} \{\frac{\bar{x}^T\bar{s} + \beta}{\bar{s}_i}, \frac{\bar{x}^T\bar{s} + \beta}{\bar{x}_i}\}$ se obtiene el resultado deseado. □

Proposición 4.1.5. *Si $\mathcal{F}^o \neq \emptyset$, entonces existe solución para el problema (P_τ).*

Demostración. Consideramos $\gamma = \text{ínf}\{f_\tau(x,s) : (x,s) \in \mathcal{A}(P_\tau)\} < +\infty$ y una sucesión minimizante

$$\{(x^l, s^l)\}_{l\in\mathbb{N}} \subset \mathcal{A}(P_\tau) \tag{4.30}$$

$$f_\tau(x^l, s^l) \to \gamma \ \text{ si } \ l \to +\infty. \tag{4.31}$$

Se pueden dar dos casos:

a) La sucesión $\{(x^l, s^l)\}_{l\in\mathbb{N}}$ está acotada. Entonces por estar en un espacio de dimensión finita podemos considerar una subsucesión convergente: $\{(x^{l'}, s^{l'})\}_{l'\in\mathbb{N}}$ tal que $(x^{l'}, s^{l'}) \to (\hat{x}, \hat{s})$ si $l' \to +\infty$. De donde se cumple que: $\tilde{A}\hat{x} = \tilde{b}, Z^T\hat{s} = Z^T c$ y además $(\hat{x}, \hat{s}) \geqslant 0$. Vamos a probar que en realidad $(\hat{x}, \hat{s}) > 0$.
Si existiera un índice i_0 tal que $\hat{x}_{i_o} = 0$, entonces tendríamos:

$$f_\tau(x^{l'}, s^{l'}) = \frac{1}{\tau}(x^{l'})^T s^{l'} - \sum_{i=1}^{n} log(x_i^{l'} s_i^{l'}) \to +\infty \ \text{ si } \ l' \to +\infty$$

porque $x_{i_o}^{l'} s_{i_o}^{l'} \to 0$ si $l' \to +\infty$. Como además $\{f_\tau(x^{l'}, s^{l'})\}_{l' \in \mathbb{N}}$ es una subsucesión de $\{f_\tau(x^l, s^l)\}_{l \in \mathbb{N}}$, usando (4.31), tendríamos que $f_\tau(x^{l'}, s^{l'}) \to \gamma$ si $l' \to +\infty$ y por la unicidad del límite llegaríamos a una contradicción $\gamma = +\infty$.
Análogamente se obtiene la misma conclusión para el caso en que hubiera una coordenada de $\hat{s}$ que se anulase.
De este modo tenemos que $(\hat{x}, \hat{s}) \in \mathcal{A}(P_\tau)$ y, como f_τ es continua, $f_\tau(x^{l'}, s^{l'}) \to f_\tau(\hat{x}, \hat{s})$ si $l' \to +\infty$.
Volviendo a usar que $\{f_\tau(x^{l'}, s^{l'})\}_{l' \in \mathbb{N}}$ es una subsucesión de $\{f(x^l, s^l)\}_{l \in \mathbb{N}}$, entonces $\gamma = f_\tau(\hat{x}, \hat{s})$. Luego $(\hat{x}, \hat{s})$ es solución de (P_τ).

b) La sucesión $\{(x^l, s^l)\}_{l \in \mathbb{N}}$ no está acotada. Entonces existe una subsucesión $\{(x^{l'}, s^{l'})\}_{l' \in \mathbb{N}}$ tal que $\|(x^{l'}, s^{l'})\| \to +\infty$ si $l' \to +\infty$. Usando el lema 4.1.4 sabemos que $\{(x^{l'})^T s^{l'}\}_{l' \in \mathbb{N}} \subset \mathbb{R}_+$ no está acotada, luego existe una subsucesión $\{(x^{l''})^T s^{l''}\}_{l'' \in \mathbb{N}}$ tal que $(x^{l''})^T s^{l''} \to +\infty$ si $l'' \to +\infty$; de donde, tomando $w^{l''} = \frac{1}{\tau}(x^{l''})^T s^{l''}$,

$$f_\tau(x^{l''}, s^{l''}) = w^{l''} - log(w^{l''}) - log(\tau) \to +\infty \quad \text{si } l'' \to +\infty.$$

Entonces razonando como hemos hecho anteriormente, llegaríamos a una contradicción al obtener que $\gamma = +\infty$.

□

Ahora probaremos la unicidad de solución siguiendo el argumento de [6].

Proposición 4.1.6. *La solución del problema (P_τ) es única.*

Demostración. Para demostrar la unicidad de solución basta aplicar el teorema 7.2.3 del anexo, teniendo en cuenta que la existencia de solución está garantizada por la proposición anterior. Vamos a ver que f_τ es estrictamente convexa en $\mathcal{A}(P_\tau)$. Para ello consideramos por separado los dos términos que definen la función.

a) $f_1(x, s) = \dfrac{1}{\tau} x^T s$

b) $f_2(x, s) = -\displaystyle\sum_{i=1}^{n} log(x_i s_i)$

Veamos que, aunque $f_1(x, s)$ es aparentemente cuadrática, restringida a $\mathcal{A}(P_\tau)$ es afín. Sea $\bar{x}$ un vector que cumple que $\tilde{A}\bar{x} = \tilde{b}$, entonces para cualquier $(x, s) \in \mathcal{A}(P_\tau)$, de $Z^T(-s + c) = 0$, como $R(\tilde{A}^T) \bigoplus N(\tilde{A}) = \mathbb{R}^n$ se tiene que $-s + c \in R(\tilde{A}^T)$ y por tanto existe λ tal que $\tilde{A}^T \lambda = -s + c$, de donde podemos deducir las siguentes igualdades.

$$x^T s = x^T(c - \tilde{A}^T\lambda) = c^T x - \tilde{b}^T\lambda = c^T x - \bar{x}^T \tilde{A}^T \lambda = c^T x - \bar{x}^T(c - s) = c^T x + \bar{x}^T s - \bar{x}^T c.$$

Por otra parte la función f_2 cumple que su segunda derivada es:

$$\nabla^2 f_2(x, s) = \begin{pmatrix} \frac{1}{x_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{x_2^2} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \frac{1}{s_n^2} \end{pmatrix}.$$

Por tanto usando el teorema 7.1.5 del anexo tenemos que la función f_2 es estrictamente convexa en $\mathcal{A}\ (P_\tau)$. Así que f_τ, por ser la suma de f_1 y f_2, es estrictamente convexa en $\mathcal{A}\ (P_\tau)$. □

Proposición 4.1.7. *Si $(\bar{x}_\tau, \bar{s}_\tau)$ es solución de (P_τ), entonces existe un único $\bar{\lambda}_\tau$ tal que $(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau)$ verifica las ecuaciones del camino central.*

Demostración. $Z^T(-\bar{s}_\tau + c) = 0$ por lo tanto como $R(\tilde{A}^T) \bigoplus N(\tilde{A}) = \mathbb{R}^n$ se tiene que $-\bar{s}_\tau + c \in R(\tilde{A}^T)$. Además usando que el rango$(\tilde{A}) = m$ existe un único $\bar{\lambda}_\tau$ tal que $\tilde{A}^T\bar{\lambda}_\tau = -\bar{s}_\tau + c$.
Probemos ahora que $(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau)$ verifica las ecuaciones del camino central. Para ello veamos que es solución del siguiente problema.

$$(\tilde{P}_\tau) \begin{cases} \text{mín} \quad \tilde{f}_\tau(x,\lambda,s) = f_\tau(x,s) \\ (x,\lambda,s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \\ \\ \tilde{A}x = \tilde{b} \\ \tilde{A}^T\lambda + s = c \\ (x,s) > 0 \end{cases} . \tag{4.32}$$

Es evidente que $(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau) \in \mathcal{A}\,(\tilde{P}_\tau)$. Dado otro punto admisible para $(\tilde{P}_\tau)$, (x,λ,s), veamos que se cumple que $\tilde{f}_\tau(x,\lambda,s) \geqslant \tilde{f}_\tau(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau)$, o lo que es lo mismo, $f_\tau(x,s) \geqslant f_\tau(\bar{x}_\tau, \bar{s}_\tau)$.
Sabemos que $(\bar{x}_\tau, \bar{s}_\tau)$ es la solución de (P_τ). Pero nos falta ver que $(x,s) \in \mathcal{A}(P_\tau)$. Como $(x,\lambda,s) \in \mathcal{A}(\tilde{P}_\tau)$, se tiene que $\tilde{A}\lambda + s = c \Rightarrow Z^T\tilde{A}\lambda + Z^Ts = Z^Tc$, y como sabemos que $\tilde{A}^TZ = 0$, obtenemos que $Z^Ts = Z^Tc$.
Ahora falta probar que $\bar{x}_i\bar{s}_i = \tau$ para $i = 1, \ldots, n$, o lo que es lo mismo, $\bar{X}_\tau\bar{S}_\tau e = \tau e$. Para ello vamos a usar las condiciones de Khun-Tucker de $(\tilde{P}_\tau)$ en $(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau)$.

$$\frac{\partial}{\partial x}\tilde{f}_\tau(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau) + \tilde{A}^T\zeta = 0 \Rightarrow \frac{\bar{s}_\tau}{\tau} - \bar{X}_\tau^{-1}e = -\tilde{A}^T\zeta, \tag{4.33}$$

$$\frac{\partial}{\partial \lambda}\tilde{f}_\tau(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau) + \tilde{A}\beta = 0 \Rightarrow 0 = -\tilde{A}\beta, \tag{4.34}$$

$$\frac{\partial}{\partial s}\tilde{f}_\tau(\bar{x}_\tau, \bar{\lambda}_\tau, \bar{s}_\tau) + \beta = 0 \Rightarrow \frac{\bar{x}_\tau}{\tau} - \bar{S}_\tau^{-1}e = -\beta, \tag{4.35}$$

$$A\bar{x}_\tau = b, \tag{4.36}$$

$$A^T\bar{\lambda}_\tau + \bar{s}_\tau = c. \tag{4.37}$$

siendo $\zeta \in \mathbb{R}^n$ y $\beta \in \mathbb{R}^n$ los multiplicadores de Lagrange correspondientes.
A continuación vamos a tomar el producto de (4.33) y (4.35),

$$(\frac{\bar{s}_\tau}{\tau} - \bar{X}_\tau^{-1}e)^T(\frac{\bar{x}_\tau}{\tau} - \bar{S}_\tau^{-1}e) = (-\tilde{A}^T\zeta)^T(-\beta)$$

y usando (4.34), obtenemos:

$$(\frac{\bar{s}_\tau}{\tau} - \bar{X}_\tau^{-1}e)^T(\frac{\bar{x}_\tau}{\tau} - \bar{S}_\tau^{-1}e) = 0.$$

Como $\bar{X}_\tau^{-\frac{1}{2}}\bar{S}_\tau^{\frac{1}{2}}\bar{X}_\tau^{\frac{1}{2}}\bar{S}_\tau^{-\frac{1}{2}} = I$, podemos proseguir de la siguiente forma sin alterar nuestra igualdad:

$$(\frac{\bar{x}_\tau}{\tau} - \bar{S}_\tau^{-1}e)^T(\bar{X}_\tau^{-\frac{1}{2}}\bar{S}_\tau^{\frac{1}{2}})(\bar{X}_\tau^{\frac{1}{2}}\bar{S}_\tau^{-\frac{1}{2}})(\frac{\bar{s}_\tau}{\tau} - \bar{X}_\tau^{-1}e)^T = 0,$$

lo que es lo mismo que,

$$\|\frac{1}{\tau}(\bar{X}_\tau\bar{S}_\tau)^{\frac{1}{2}}e - (\bar{S}_\tau\bar{X}_\tau)^{-\frac{1}{2}}e\|_2^2 = 0,$$

de donde

$$\frac{1}{\tau}(\bar{X}_\tau\bar{S}_\tau)^{\frac{1}{2}}e - (\bar{S}_\tau\bar{X}_\tau)^{-\frac{1}{2}}e = 0$$

o equivalentemente,

$$(\bar{S}_\tau\bar{X}_\tau)^{\frac{1}{2}}(\bar{X}_\tau\bar{S}_\tau)^{\frac{1}{2}}e - \tau e = 0$$

obteniendo lo que estábamos buscando,

$$\bar{S}_\tau\bar{X}_\tau = \tau e.$$

□

LOS ENTORNOS DEL CAMINO CENTRAL.

Los métodos que siguen el camino central restringen los iterantes a un entorno del mismo. Los dos entornos más frecuentemente utilizados son:

$N_2(\theta) = \{(x, \lambda, s) \in \mathcal{F}^o \text{ tal que } \|XSe - \mu e\|_2 \leqslant \theta\mu\}$ para algún $\theta \in [0, 1)$.
$\mathcal{N}_{-\infty}(\gamma) = \{(x, \lambda, s) \in \mathcal{F}^o \text{ tal que } x_i s_i \geqslant \gamma\mu \text{ para todo } i = 1, 2, ..., n\}$ para algún $\gamma \in (0, 1]$.

Valores típicos de los parámetros son $\theta = 0{,}5$ y $\gamma = 10^{-3}$.
El entorno $\mathcal{N}_2(\theta)$ es más restrictivo que $\mathcal{N}_{-\infty}(\gamma)$ como veremos en el siguiente resultado.

Proposición 4.1.8. *Sea $\theta \in (0, 1)$. Se cumple $\mathcal{N}_2(\theta) \subset \mathcal{N}_{-\infty}(\gamma)$ para $0 < \gamma \leqslant 1 - \theta$.*

Demostración. Si $(x, \lambda, s) \in \mathcal{N}_2(\theta)$, tenemos que: $\sum_{i=1}^{n}(x_i s_i - \mu)^2 \leqslant (\theta\mu)^2$. De donde, usando que $\sum_{i=1}^{n}(x_i s_i - \mu)^2 \geqslant (x_j s_j - \mu)^2$ para cualquier $j \in \{1, ..., n\}$, $(x_j s_j - \mu)^2 \leqslant (\theta\mu)^2$ para cualquier $j \in \{1, ..., n\}$. Es decir que $|x_j s_j - \mu| \leqslant \theta\mu$ de donde $-\theta\mu \leqslant x_j s_j - \mu \leqslant \theta\mu$ y, fijándonos en la cota inferior, tenemos que $-\theta\mu + \mu \leqslant x_j s_j$ o, lo que es lo mismo, $\mu(1 - \theta) \leqslant x_j s_j$ y, por tanto, $\mu\gamma \leqslant x_j s_j$, para $\gamma \in (0, 1 - \theta)$. □

Capítulo 5

Experimentación numérica

En la experimentación numérica realizada se han considerado dos tipos de problemas, unos generados aleatoriamente a los que dedicamos la primera sección de este capítulo y otros relacionados con la recuperación de señales, cuyos resultados numéricos aparecen en la segunda sección. En ambos casos, los resultados presentados se han obtenido con los códigos linprog de MATLAB y l1eq_pd de l_1- MAGIC y las tolerancias son respectivamente $\bar{\epsilon} = 10^{-8}$ y $\epsilon = 10^{-3}$. En el segundo caso no ha sido posible tomar una tolerancia menor y obtener una solución satisfactoria.

5.1. Recuperando soluciones sparse

PRIMER EXPERIMENTO: equivalencia de (P_0) y (P_1)

Hemos llevado a cabo un experimento inspirado en [3] que involucra la generación de varios problemas aleatorios para analizar en la práctica si resolviendo (P_1) obtenemos una solución para (P_0). Para ello hemos construido cien matrices aleatorias, $A \in \mathbb{R}^{100\times 200}$, siguiendo la distribución $\mathcal{N}(0,1)$. Para todas ellas se cumple que spark(A)=101 con probabilidad 1 (ver [3]), por lo que, según el teorema 3.0.2, dado $b \in \mathbb{R}^{100}$, toda solución del sistema $Ax = b$ que tenga menos de 51 coordenadas distintas de cero, es necesariamente la más sparse posible y por tanto la solución del problema (P_0). Para cada matriz A, hemos generado 70 vectores sparse, $\tilde{x}$, con un número de coordenadas distintas de cero variando de 1 a 70. Estas coordenadas han tomado valores aleatorios siguiendo la distribución $\mathcal{N}(0,1)$ que han sido colocados uniformemente en cada vector $\tilde{x}$. Una vez generado el adecuado término independiente, $b = A\tilde{x}$, $\tilde{x}$ es solución del correspondiente sistema, $Ax = b$.

Cada uno de los 7000 problemas planteados ha sido resuelto con linprog de MATLAB para las formulaciones $(\tilde{P}_1)$ y $(\tilde{\tilde{P}}_1)$ y con l1eq_pd de l_1- MAGIC para la formulación $(\tilde{P}_1)$. Para cada valor $||\tilde{x}||_0$ hemos calculado la tasa de éxito al resolver los 100 problemas asociados. En los casos en los que $||\tilde{x}||_0 \geq 51$ hemos considerado éxito cuando la solución calculada tiene un número de coordenadas distintas de cero igual o menor que el correspondiente valor $||\tilde{x}||_0$.

Empezamos por comparar el comportamiento de linprog con las dos formulaciones $(\tilde{P}_1)$ y $(\tilde{\tilde{P}}_1)$. En la figura 5.1 podemos observar que los resultados son mejores con la fomulación $(\tilde{\tilde{P}}_1)$ por lo tanto es la que usaremos para el resto de las figuras cuando se haya trabajado con linprog. Los resultados de l_1-MAGIC corresponden a la formulación $(\tilde{P}_1)$ (usada internamente en l1eq_pd).

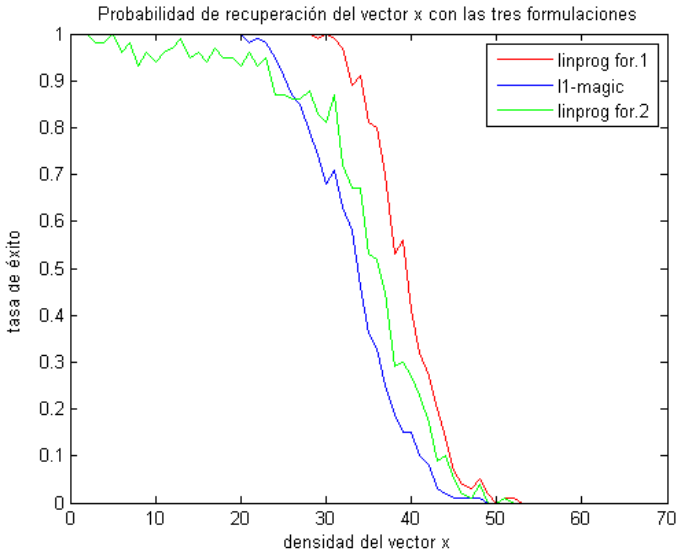

Figura 5.1: Tasa de éxito en función del número de coordenadas no nulas del vector a recuperar. En rojo los resultados obtenidos con la función linprog para la formulación $(\tilde{\tilde{P}}_1)$, en verde los correspondiente a la función linprog para la formulación $(\tilde{P}_1)$ y en azul los de la función l1eq_pd de l_1-MAGIC

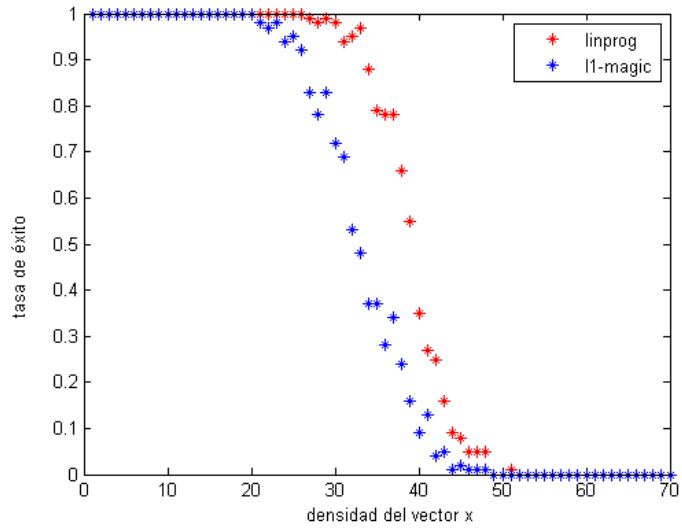

Figura 5.2: Tasa de éxito en función del número de coordenadas no nulas del vector a recuperar. En rojo los resultados obtenidos con la función linprog de MATLAB y en azul los correspondientes a la función l1eq_pd de l_1-MAGIC

En la figura 5.2 se compara el comportamiento de linprog y l_1-MAGIC al resolver los 7000 problemas propuestos. Para cada uno de los dos códigos se muestra la tasa de éxito que hemos obtenido en función de la densidad del vector a recuperar. Podemos observar que se obtienen mejores resultados con la función linprog de MATLAB, lo que podría estar motivado porque l_1-MAGIC es un software experimental. En los rangos extremos, para $1 \leq \|x\|_0 \leq 20$ y $\|x\|_0 \geq 52$, ambos códigos se comportan igual. Para la función linprog el éxito de recuperación de la solución más sparse posible (esto es resolviendo (P_1), resolvemos (P_0)) está garantizado cuando

$1 \leqslant \|x\|_0 \leqslant 35$. Para los vectores con más de 50 coordenadas no nulas, es decir, los que cumplen que $\|x\|_0 > \frac{spark(A)}{2}$, la tasa de éxito es nula (no se resuelve (P_0)).
A continuación, en la tabla siguiente, presentamos algunos resultados numéricos para el caso especial en el que las matrices aleatorias consideradas anteriormente son tales que el número de columnas duplica al número de filas y que avalan la afirmación de [10] en la que se dice que la equivalencia de (P_0) y (P_1) se mantiene para soluciones con un nivel de densidad menor que ρn donde $\rho > 0$ es una constante independiente de las dimensiones de la matriz. En la tabla, para cada tamaño de matriz considerado, se presenta la cota superior experimental de acierto (CSEA), que indica que a partir de ella la tasa de éxito resolviendo 10 problemas asociados al mismo tipo de densidad ha dejado de ser 1, y además se muestra el valor $CSEA/n$ en la última columna.

Tamaño de la matriz	**CSEA**	ρ
m=25; n=50	5	0.1
m=50; n=100	13	0.1
m=100; n=200	29	0.15
m=150; n=300	47	0.16
m=200; n=400	58	0.15
m=250; n=500	85	0.17

CSEA: Cota superior experimental de acierto. A partir de esta cota la tasa de éxito deja de ser 1.

SEGUNDO EXPERIMENTO: recuperación de soluciones sparse cuando la matriz A está mal equilibrada. Influencia del escalamiento

En este experimento nos hemos interesado en la recuperación de soluciones sparse de problemas cuya matriz del sistema no está bien equilibrada. Esto viene motivado por el diferente comportamiento de las funciones objetivo de los problemas (P_0) y (P_1): mientras $\|x\|_0$ no se ve afectada por el tamaño de las coordenadas del vector x, la norma $\|\cdot\|_1$ tiende a penalizar los valores más grandes de las coordenadas del vector y así proporcionar soluciones con coordenadas distintas de cero asociadas a columnas de la matriz con normas grandes. Es por ello que nos hemos planteado la siguiente formulación, que introduce un escalamiento en las variables en la función objetivo.

$$\begin{cases} \text{mín } \|Wx\|_1 \\ x \in \mathbb{R}^{200} \\ \text{sujeto a } Ax = b \end{cases}, \tag{5.1}$$

donde W es una matriz diagonal tal que: $W_{ii} = \|(A_{1i}, A_{2i}, \ldots, A_{mi})\|_2$ para $1 \leqslant i \leqslant n$.
En este experimento hemos generado problemas aleatorios de igual forma que en el primero de esta sección, modificando las matrices una vez generadas: multiplicando a las columnas pares de A por 10. Para cada problema hemos resuelto dos formulaciones: $\tilde{\tilde{P}}_1$ y su versión escalada siguiendo (5.1), ambas con la función linprog de MATLAB. Al igual que en el primer experimento dibujamos la tasa de éxito en función del número de coordenadas no nulas del vector a recuperar. En este caso la densidad de los vectores solución es menor que 50 y para cada matriz se han generado 10 problemas, obteniéndose un total de 500 problemas distintos. Cada uno de ellos se ha resuelto dos veces, una sin modificación y otra con ella. Solamente cinco de las 1000 ejecuciones pararon por haber saturado el número máximo de iteraciones permitido. Para esos problemas, en la siguiente tabla se presentan resultados numéricos asociados a las condiciones de optimalidad. Vemos que, a pesar de haber saturado las iteraciones, el proceso de optimización se ha realizado satisfactoriamente en todos los casos. En la tabla se asigna un 1 a las dificultades aparecidas al resolver la formulación sin escalar y un 2 a la formulación (5.1).

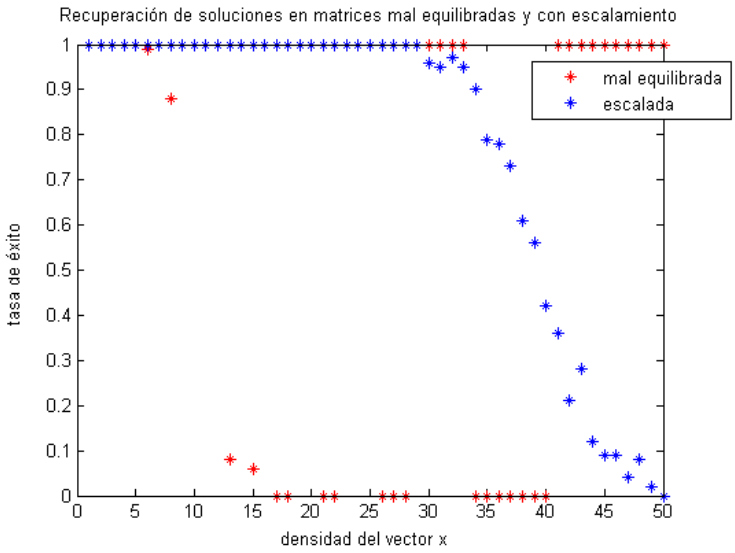

Figura 5.3: Tasa de éxito en función del número de coordenadas no nulas del vector a recuperar. Los asteriscos rojos corresponden a los casos con la matriz mal equilibrada, mientras que los azules están asociados a los problemas escalados.

Dificultades en:	**Densidad del vector** x	**constrviolation**	**firstorderopt**	**iteraciones**
Dificultades en 1	8	1.5557 e-05	8.0950 e-07	85
Dificultades en 1	8	3.3041 e-06	1.6331 e-08	85
Dificultades en 2	38	7.1842 e-06	1.7940 e-08	85
Dificultades en 2	37	7.7831 e-08	2.3962 e-06	85
Dificultades en 2	38	3.7467 e-07	2.4958 e-05	85

Constrviolation: violación de las restricciones. **Firstorderopt**: en relación a las condiciones de optimalidad de primer orden (7.3) y (7.4) del anexo.

En la figura 5.3 se observa como para valores de densidad menores que 40, con la matriz mal equilibrada no se recupera la solución de (P_0) en un 40 % de los casos. Mientras tanto, con la mejora introducida, la tasa de éxito al recuperar la solución de (P_0) sigue siendo 1 cuando el vector x tiene densidad menor que 30. Con el problema no escalado el primer fallo lo obtenemos con un nivel de densidad muy pequeño, 6. Por otro lado se observa que para niveles grandes de densidad (para valores mayores que 40) el escalamiento funciona peor que la versión no escalada.

5.2. Recuperando señales

Abordamos ahora el problema de la recuperación de señales. Siguiendo la referencia [2], hemos trabajado con una señal que se corresponde con el sonido de un tono telefónico, que viene descrita por la suma de dos sinusoides $f(t) = sen(1349\pi t) + sen(3266\pi t)$ y que en nuestros experimentos viene dada por un vector, $\hat{f}$, con un total de 5000 coordenadas. Podemos representar $\hat{f}$ como combinación lineal de cierta base: $\hat{f} = \Psi c$ donde Ψ representa la DCT (transformada discreta del coseno) y c son los coeficientes. La transformada discreta del coseno, es usada en el procesamiento de señales e imágenes porque tiene la propiedad de "compactar energía": en las aplicaciones típicas, la mayoría de la información de la señal tiende a estar concentrada en unas pocas

componentes de pequeña frecuencia de la DCT. Es similar a la transformada de Fourier discreta pero la diferencia es que sólo usa números reales. La DCT se utiliza en el formato JPEG para comprimir imágenes (ver [8]).

En nuestro experimento la señal comprimida es un vector, b, con m muestras aleatorias de $\hat{f}$. Luego $b = \Phi\hat{f}$, siendo Φ un subconjunto de las filas de la matriz identidad. Por lo tanto para reconstruir la señal hay que recuperar los coeficientes resolviendo el siguiente sistema de ecuaciones: $Ax = b$, donde $A = \Phi\Psi$. Una vez hemos calculado numéricamente los coeficientes, x, Ψx determinará la señal recuperada. El primer gráfico de la siguiente figura muestra el comienzo de nuestra señal original $\hat{f}$ y las correspondientes observaciones.

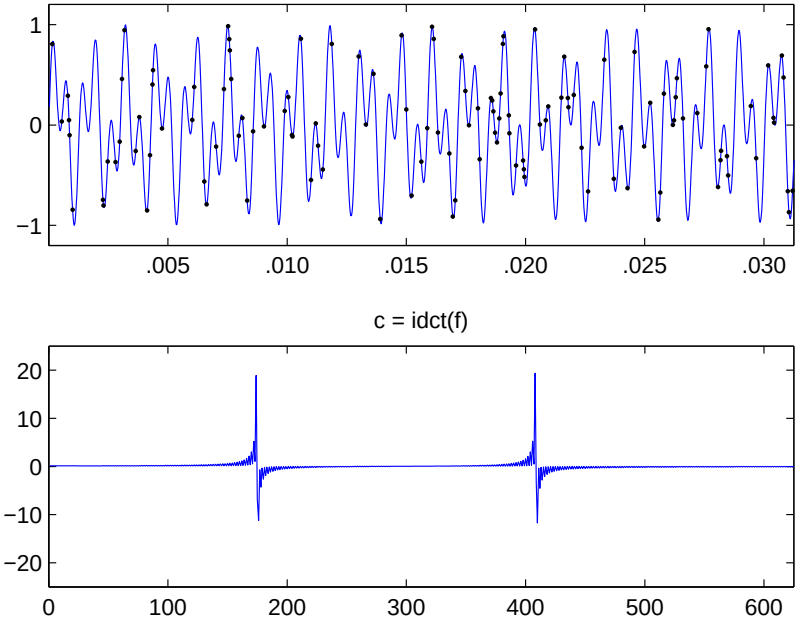

Figura 5.4: Señal original y observaciones aleatorias arriba y coeficientes originales debajo

El segundo gráfico de la figura 5.4 muestra los primeros coeficientes. Los coeficientes no mostrados son de un tamaño despreciable respecto a los de la figura. Dado que nuestro vector de observaciones (b) está formado por 500 componentes, la matriz (A) está formada por 500 filas y 5000 columnas por lo que el sistema de ecuaciones resultante: $Ax = b$ tiene 10 veces más variables que ecuaciones.

PRIMER EXPERIMENTO: recuperación con el software l_1- MAGIC.

Para el problema planteado hemos resuelto las dos formulaciones (P_1) y (P_2) con el objetivo de mostrar que (P_1) es más adecuado que (P_2) para la aplicación que nos ocupa.

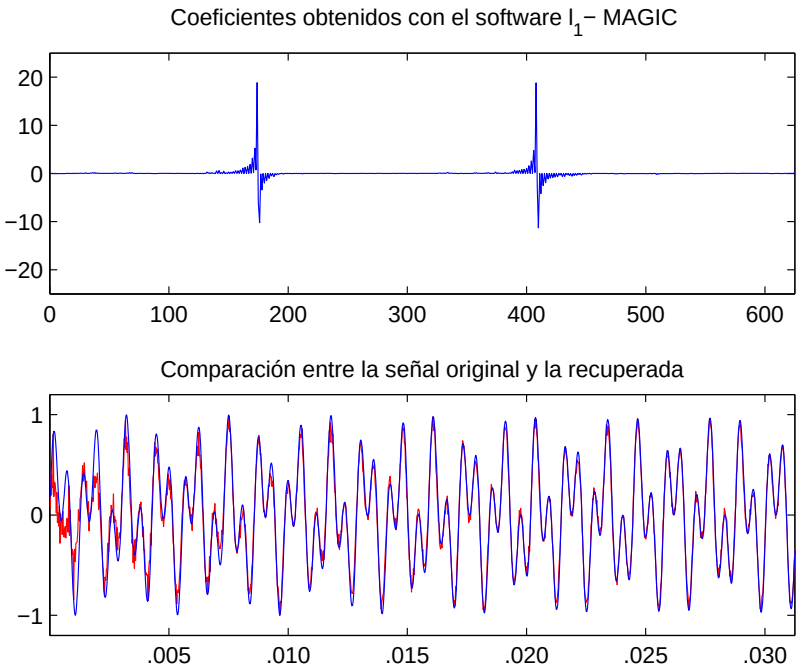

Figura 5.5: Recuperación con l_1- MAGIC

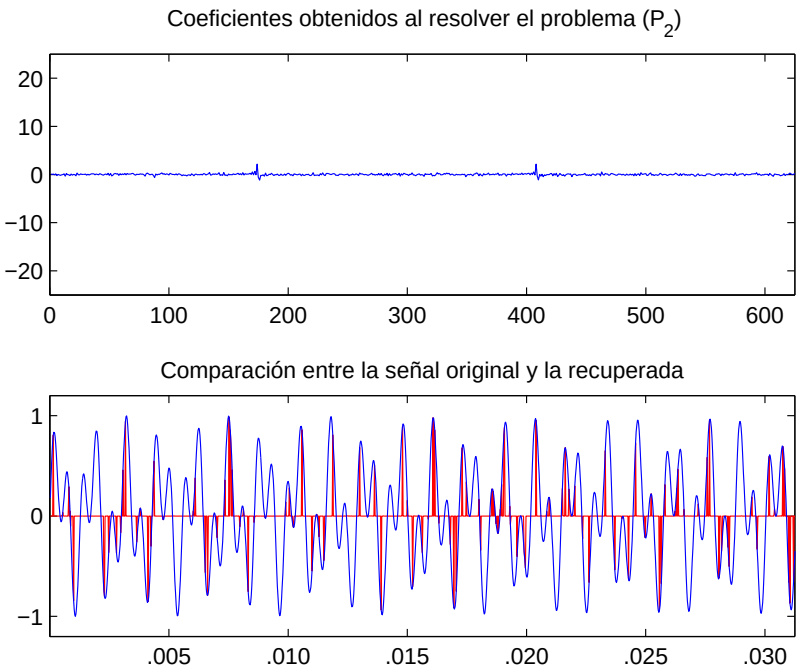

Figura 5.6: Recuperación con la norma 2

En el primer gráfico de la figura 5.5 se muestran los coeficientes con mayor peso de la solución obtenida para el problema (P_1). Estos se asemejan bastante a los coeficientes originales (ver figura 5.4). En el segundo gráfico, donde se compara la señal numérica obtenida con la señal original, observamos que en la primera parte la recuperación no ha sido tan positiva como en el

resto.
En la figura 5.6 se muestran los resultados obtenidos con la resolución del problema de optimización (P_2). Podemos observar que la recuperación de la señal tiene mucha menos calidad que la obtenida en la figura 5.5, asociada a la formulación (P_1).

SEGUNDO EXPERIMENTO: variando el número de observaciones.

Vamos a comprobar como varía la recuperación de la señal si tomamos muchas más observaciones. Para ello hemos tomado las 2500 observaciones aleatorias (las primeras son mostradas en la figura 5.7). Para poder observar mejor la recuperación de la señal y compararla con la original la hemos dibujado en cinco tramos (ver figuras 5.8, 5.9 y 5.10). Aparece en rojo la señal original y en verde la recuperada. Observamos que en la mayoría de los tramos la diferencia entre ambas señales es inapreciable a simple vista. En este caso hemos usado el software l_1-MAGIC.

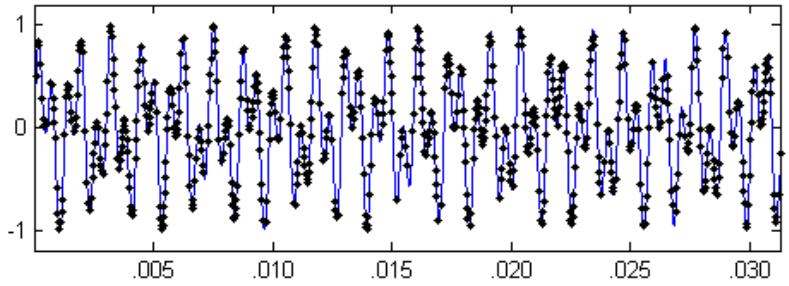

Figura 5.7: Señal original y las 2500 observaciones

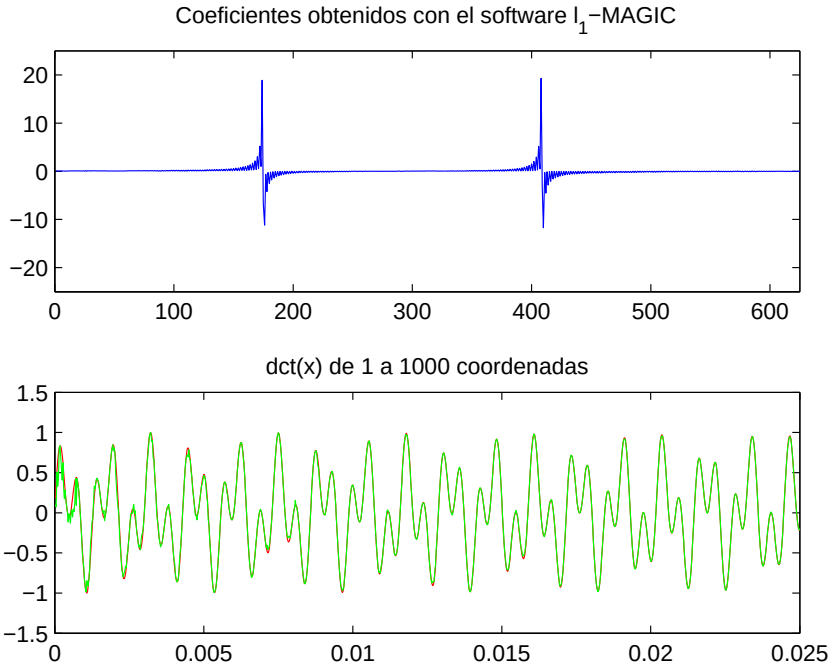

Figura 5.8: Coeficientes y señal recuperada de 1 a 1000 coordenadas

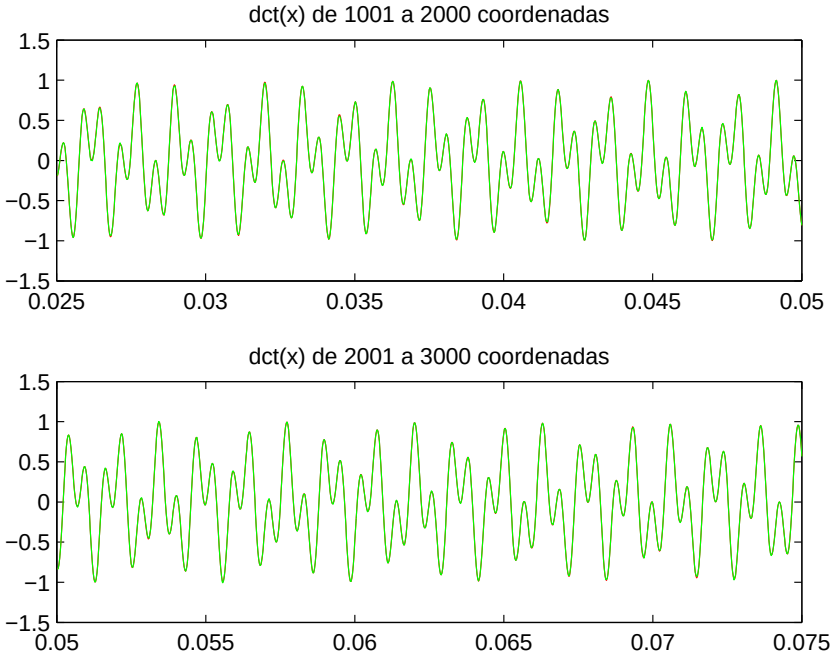

Figura 5.9: Señal recuperada de 1001 a 2000 y de 2001 a 3000

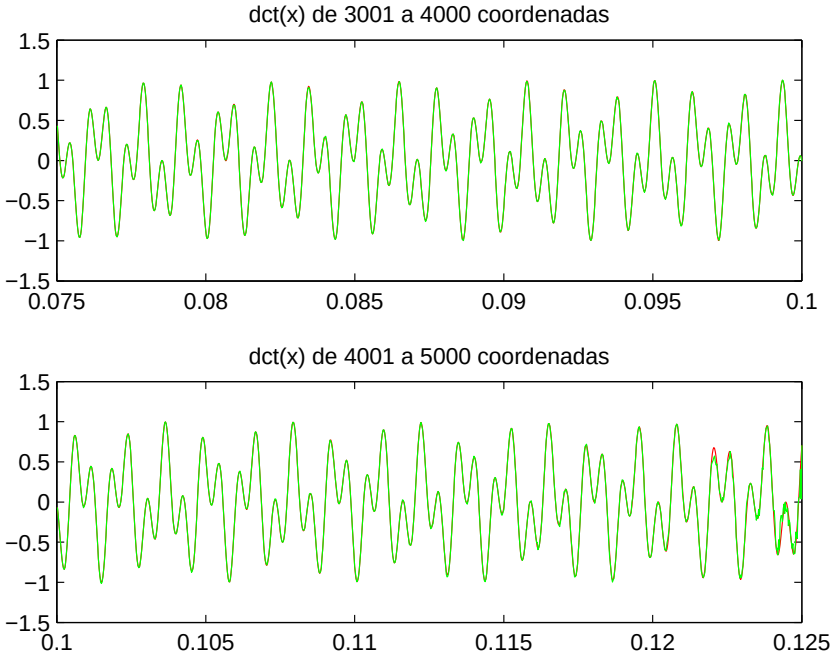

Figura 5.10: Señal recuperada de 3001 a 4000 y de 4001 a 5000

A continuación hemos calculado el error relativo de la señal recuperada frente a la original en estos cinco tramos y los resultados obtenidos son los que aparecen en la siguiente tabla. Para cada tramo, en la segunda columna se muestra el error obtenido a partir de las 500 observaciones aleatorias y en la última columna el error correspondiente a haber tomado las 2500 observaciones

aleatorias.

Tramo	E.r. con m=500	E.r. con m=2500
De 1 a 1000	$2{,}6022e-01$	$7{,}6981e-02$
De 1001 a 2000	$8{,}5807e-02$	$1{,}3856e-02$
De 2001 a 3000	$6{,}7361e-02$	$1{,}0686e-02$
De 3001 a 4000	$7{,}4543e-02$	$1{,}3248e-02$
De 4001 a 5000	$1{,}9969e-01$	$6{,}3198e-02$

Como conclusión podemos decir que los errores más pequeños se cometen entre las coordenadas 1001 a 4000 en ambos experimentos. Además, como era de esperar, se observa una disminución del error relativo en el experimento al aumentar el número de observaciones.
Por último veamos si mejora la recuperación de la señal resolviendo el problema (P_2) al aumentar el número de observaciones a 2500.

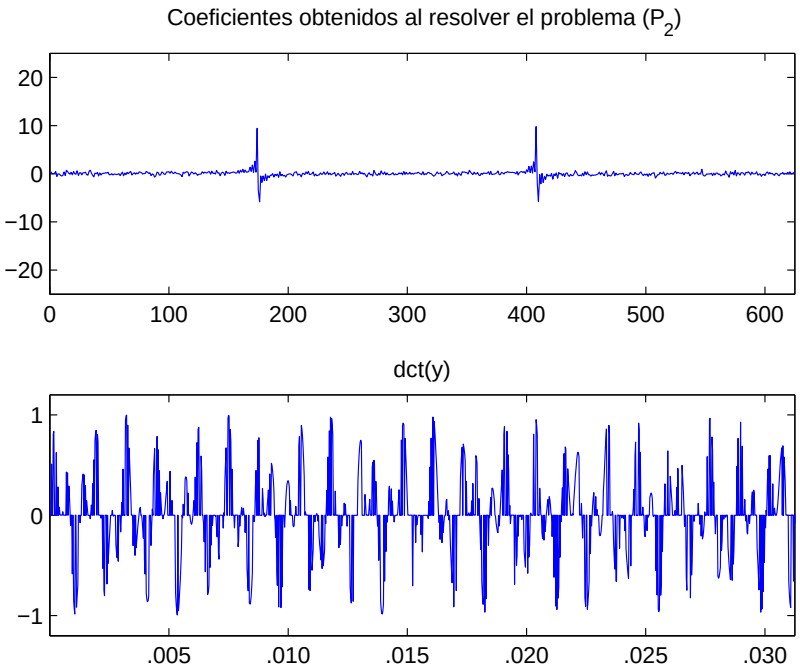

Figura 5.11: Recuperación de la señal con la norma 2

Observamos que la solución obtenida dista mucho de tener una calidad aceptable para el objetivo planteado.

TERCER EXPERIMENTO: recuperación con la formulación $(\tilde{P}_1)$ y el código linprog

Nuestro tercer experimento consiste en recuperar la señal pero usando el código linprog y la formulación del problema $(\tilde{P}_1)$ (ver 2.5).
Hemos repetido el primer experimento de esta sección (con 500 observaciones).
Al igual que antes hemos dibujado la señal recuperada en cinco secciones para poder compararla mejor. Por lo tanto en los siguientes gráficos van a aparecer los coeficientes obtenidos y la gráfica recuperada en primer lugar de 1 a 1000 coordenadas, luego de 1001 a 2000, de 2001 a 3000, de 3001 a 4000 y de 4001 a 5000 respectivamente. Como anteriormente se ha hecho, la señal original está dibujada en rojo y la recuperada con linprog en verde.

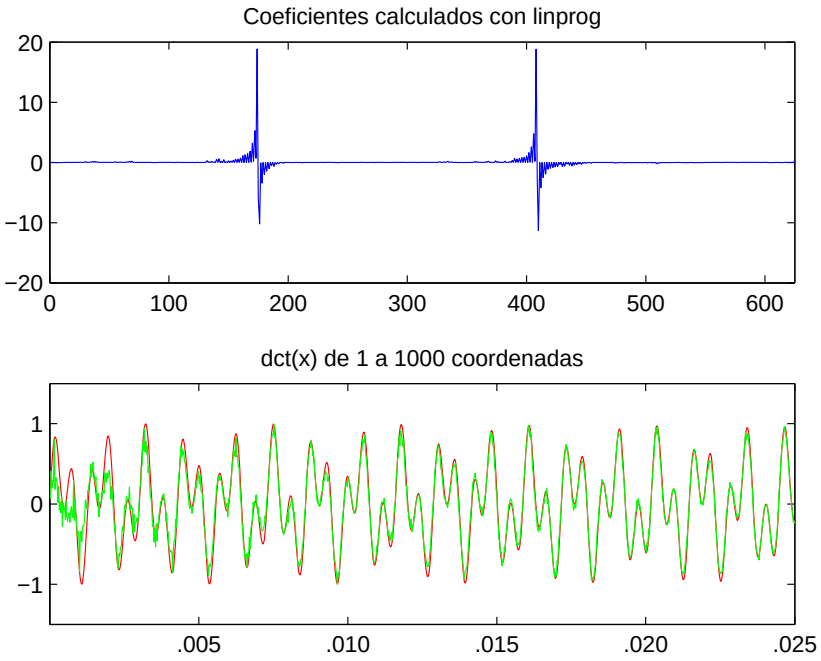

Figura 5.12: Coeficientes y señal recuperada de 1 a 1000

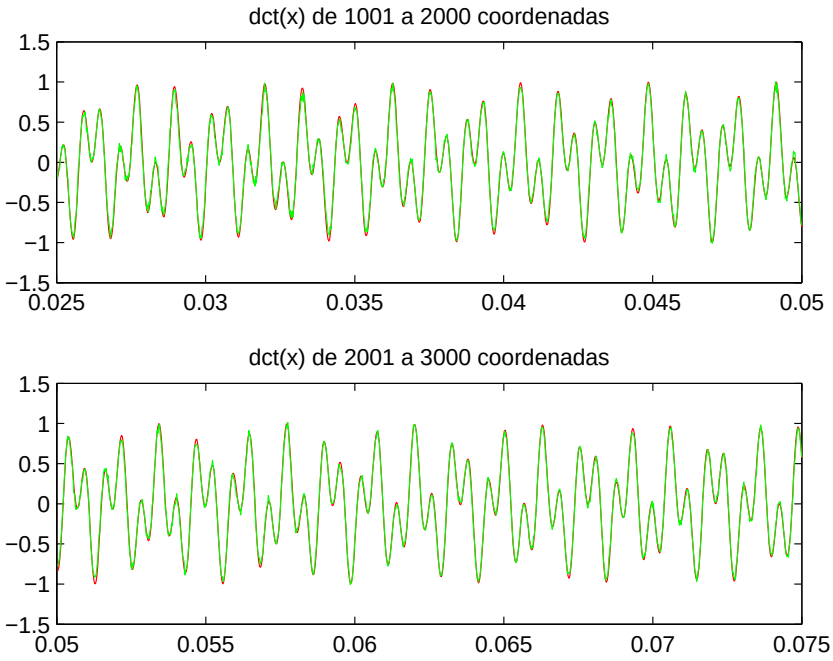

Figura 5.13: Señal recuperada de 1001 a 2000 y de 2001 a 3000

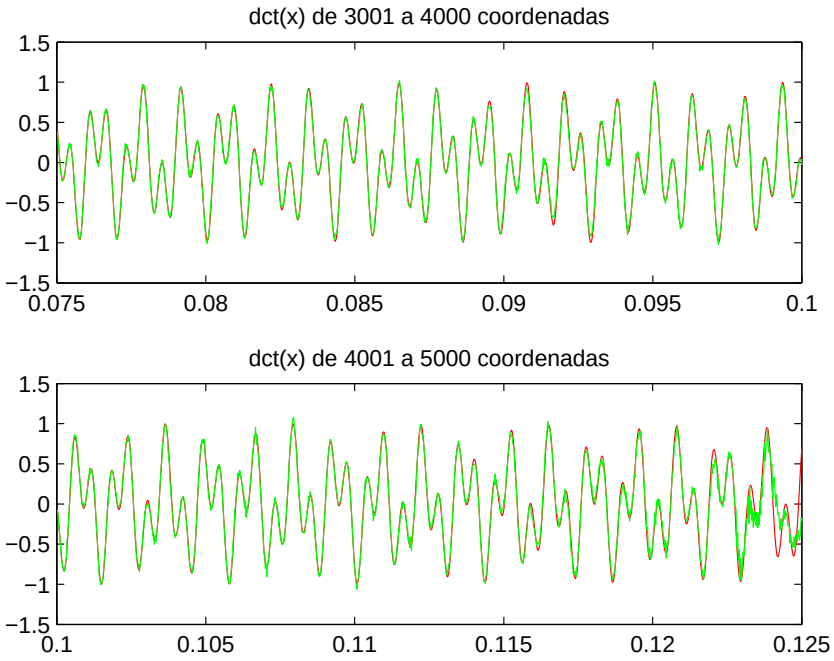

Figura 5.14: Señal recuperada de 3001 a 4000 y de 4001 a 5000

Se puede observar a simple vista que en la primera sección, de 1 a 1001 coordenadas, la recuperación es peor que en el resto de las secciones. Para verlo objetivamente también hemos analizado los errores relativos en las cinco secciones obteniendo los resultados que se presentan en la siguiente tabla. En este caso los errores más pequeños también se cometen entre los coeficientes 1001 y 4000 y son similares a los obtenidos con el software l_1-MAGIC.

Tramo	E.r. con m=500
De 1 a 1000	$2{,}6027e-01$
De 1001 a 2000	$8{,}5698e-02$
De 2001 a 3000	$6{,}7442e-02$
De 3001 a 4000	$7{,}4462e-02$
De 4001 a 5000	$1{,}9966e-01$

CUARTO EXPERIMENTO: variando la distribución de las observaciones.

En este experimento hemos tomado las 500 observaciones equiespaciadas (las primeras son mostradas en la figura 5.15), en vez de tomar muestras aleatorias como hemos hecho anteriormente, con el objetivo de ver cómo influye este cambio en la recuperación de la señal.

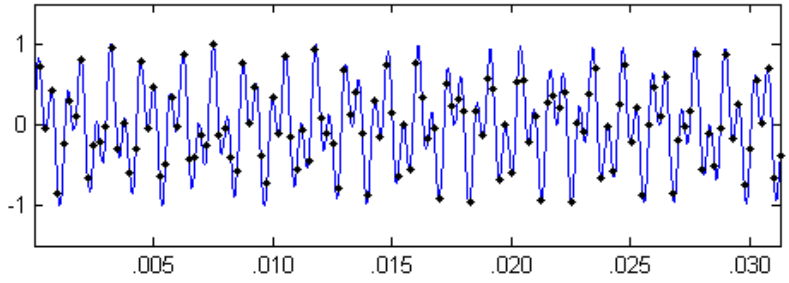

Figura 5.15: Señal original y observaciones equiespaciadas

En la figura 5.16 mostramos la recuperación conseguida con l_1- MAGIC. Podemos observar el gran cambio que se produce el tomar las observaciones equiespaciadas frente a las aleatorias, pues la recuperación es mucho peor. Los coeficientes obtenidos tienen unos picos muy pequeños frente a los originales y la señal recuperada (en verde) muestra grandes diferencias frente a la señal original (en rojo).

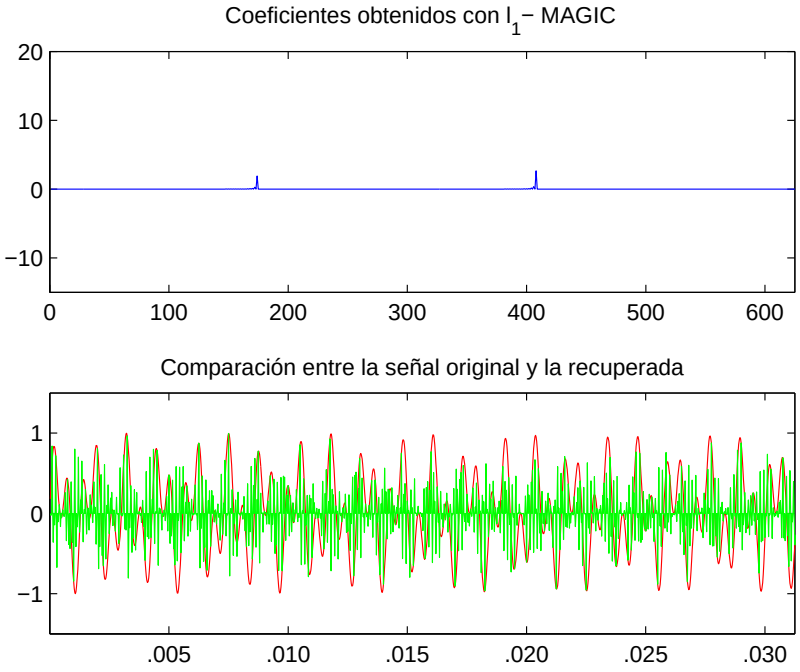

Figura 5.16: Coeficientes obtenidos y señal recuperada con la norma 1

En la siguiente tabla, en la segunda y la tercera columna, se muestran los errores relativos en las cinco secciones. La segunda corresponde a las 500 observaciones y la tercera a las 2500.

Tramo	E.r. m=500	E.r. m=2500
De 1 a 1000	$1{,}0913e+00$	$7{,}0714e-01$
De 1001 a 2000	$1{,}0533e+00$	$7{,}0757e-01$
De 2001 a 3000	$1{,}0417e+00$	$7{,}0680e-01$
De 3001 a 4000	$1{,}0429e+00$	$7{,}0693e-01$
De 4001 a 5000	$1{,}0648e+00$	$7{,}0753e-01$

Podemos observar que incluso tomando 2500 observaciones equiespaciadas no se alcanza la calidad de los resultados obtenidos al tomar 500 de forma aleatoria.
Por último, en la figura 5.17, vemos la recuperación obtenida al resolver el problema (P_2) cuando las observaciones son equiespaciadas. Este resultado es aún peor que cuando elegimos las observaciones aleatoriamente como era de esperar.

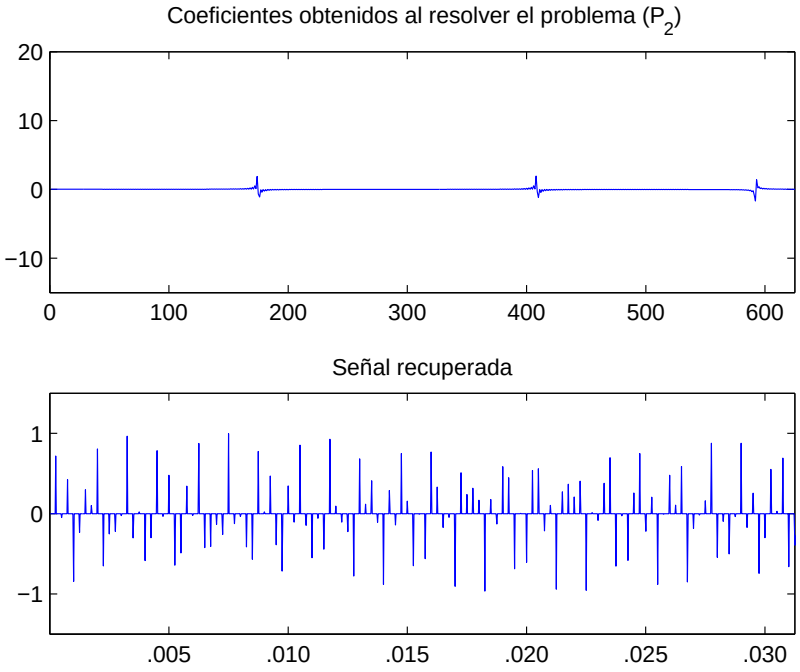

Figura 5.17: Coeficientes obtenidos y señal recuperada con la norma 2

Capítulo 6

Conclusiones

En este trabajo hemos partido de un problema clásico de álgebra lineal, resolver sistemas de ecuaciones lineales, para el que, en principio, parece que es difícil aportar algo novedoso hoy en día. Sin embargo, recientemente ha surgido un gran interés, tanto teórico como práctico, por las soluciones sparse de sistemas de ecuaciones lineales indeterminados. Se comienza tratando de minimizar la función $\|\cdot\|_0$, es decir, la función que cuenta el número de coordenadas distintas de cero que tiene un vector. Esta es la forma más intuitiva de encontrar el vector más sparse, sin embargo esta función no es continua en general y da lugar a un problema combinatorio cuya complejidad computacional hace inviable su resolución práctica. Es por esta razón que se consideran formulaciones alternativas: minimizar la norma $\|\cdot\|_1$, pues es una relajación convexa de la norma $\|\cdot\|_0$. Sin embargo está función no es diferenciable y por tanto se usan formulaciones asociadas a este problema que si lo son. Consideramos dos formulaciones que, a pesar de tener el doble de variables que el problema de partida, son más viables de resolver en la práctica, al tratarse de problemas de programación lineal. Además puede considerarse la minimización de la norma $\|\cdot\|_2$, formulación atractiva porque tiene unicidad de solución, que puede ser calculada con la SVD de la matriz del sistema. Sin embargo en la práctica con esta formulación no se recupera una solución sparse.
Para resolver problemas de programación lineal de talla grande conviene recurrir a los métodos de puntos interiores y en particular, los métodos primales-duales.
En la experimentación numérica llevada a cabo en el trabajo, de los dos códigos utilizados, podemos concluir que la mejor recuperación de la solución se consigue con la función linprog de MATLAB, para la formulación de programación lineal estándar dada en el trabajo. Además hemos comprobado que para cierto tipo de matrices, cuanto más sparse son las soluciones, mejores resultados se obtienen. Por otro lado, el escalamiento mejora mucho la recuperación cuando la matriz de nuestro sistema no está equilibrada. Finalmente, tomar muestras aleatorias mejorará notablemente los resultados numéricos.

Capítulo 7

Anexo

7.1. Condiciones de optimalidad

Sea el siguiente problema:

$$(PNL)\begin{cases} \text{Minimizar} \quad f(x) \\ \text{sujeto a} \quad x \in \Omega \subset \mathbb{R}^n, \\ h_i(x) = 0, \quad 1 \leqslant i \leqslant n_I, \\ g_j(x) \leq 0, \quad 1 \leqslant j \leqslant n_D, \end{cases} \tag{7.1}$$

con $f, h_i, g_j : \Omega \to \mathbb{R}$, siendo Ω un abierto de $\mathbb{R}^n$.

Teorema 7.1.1. ***Condiciones de optimalidad necesarias de primer orden.***
Sean $f, h_i, g_j : \Omega \to \mathbb{R}$ funciones de clase C^1, para $1 \leqslant i \leqslant n_I, 1 \leqslant j \leqslant n_D$.
Si $\bar{x} \in \Omega$ es una solución del problema PNL, entonces existen $1 + n_I + n_D$ números: $\bar{\alpha} \in \mathbb{R}_+, \{\bar{\lambda}_i\}_{i=1}^{n_I} \subset \mathbb{R}$ y $\{\bar{\mu}_j\}_{j=1}^{n_D} \subset \mathbb{R}_+$ tales que

$$\bar{\alpha} + \sum_{i=1}^{n_I} |\bar{\lambda}_i| + \sum_{j=1}^{n_D} \bar{\mu}_j > 0 \tag{7.2}$$

$$\bar{\alpha}\nabla f(\bar{x}) + \sum_{i=1}^{n_I} \bar{\lambda}_i \nabla h_i(\bar{x}) + \sum_{j=1}^{n_D} \bar{\mu}_j \nabla g_j(\bar{x}) = 0 \tag{7.3}$$

$$\bar{\mu}_j \geqslant 0 \quad y \quad \bar{\mu}_j g_j(\bar{x}) = 0, \qquad 1 \leqslant j \leqslant n_D \tag{7.4}$$

$$h_i(\bar{x}) = 0, \quad 1 \leqslant i \leqslant n_I, \qquad g_j(\bar{x}) \leqslant 0, \quad 1 \leqslant j \leqslant n_D. \tag{7.5}$$

Corolario 7.1.2. *En el resultado anterior puede tomarse $\bar{\alpha} = 1$, si además de las hipótesis del teorema anterior, se verifica una de las dos siguientes condiciones:*

a) Si $\{\nabla h_i(\bar{x})\}_{i=1}^{n_I} \cup \{\nabla g_j(\bar{x})\}_{j \in J}$, siendo $J = \{j \in \{1, ..., n_D\} : g_j(\bar{x}) = 0 \;\; y \;\; \bar{\mu_j} > 0\}$, son linealmente independientes.

b) Si las restricciones de PNL son lineales.

Definición 7.1.3. *Cuando las condiciones de optimalidad necesarias de primer orden se escriben con $\bar{\alpha} = 1$, se denominan* ***condiciones de Kuhn-Tucker****. Si $\bar{x}$ verifica estas últimas se dice que $\bar{x}$ es un* ***punto de Kuhn-Tucker****.*

Teorema 7.1.4. ***Condiciones de optimalidad suficientes de primer orden***
Sean $f, h_i, g_j : \Omega \to \mathbb{R}$ funciones de clase C^1, para $1 \leqslant i \leqslant n_I$ y $1 \leqslant j \leqslant n_D$ tales que:

- *$K = \{x \in \Omega : h_i(x) = 0$ para $1 \leqslant i \leqslant n_I$; $g_j(x) \leqslant 0$ para $1 \leqslant j \leqslant n_D\}$ es un conjunto convexo.*

– *f es una función convexa en K.*

Si $\bar{x} \in K$ es un punto de Kuhn-Tucker para el problema (PNL), entonces $\bar{x}$ es solución global del problema.

Teorema 7.1.5. *Sea $f : \Omega \longrightarrow \mathbb{R}$ de clase C^2 y $K \subset \Omega$ un conjunto convexo. Si se cumple que $(y-x)^T \nabla^2 f(x)(y-x) > 0, \forall x, y \in K, x \neq y$, entonces f es estrictamente convexa sobre K.*

7.2. Resultados de existencia de solución

Proposición 7.2.1. *Sea $K \subset \mathbb{R}^n, K \neq \emptyset$, $f : K \to \mathbb{R}$ y el problema:*

$$(P) \begin{cases} \textit{Minimizar} & f(x) \\ \textit{sujeto a} & x \in K. \end{cases} \tag{7.6}$$

Si f es coerciva en K, continua y K es cerrado, entonces existe al menos una solución (global) para (P).

Proposición 7.2.2. *Si K es un conjunto convexo y f es una función convexa en K, entonces toda solución local de (P) es solución global de (P).*

Teorema 7.2.3. *Si existe solución para (P), K es un conjunto convexo y f es una función estrictamente convexa en K, entonces el problema (P) admite una única solución (global).*

Proposición 7.2.4. *Sea el siguiente problema de programación lineal:*

$$(PL) \begin{cases} \textit{Minimizar} & f(x) = c^T x \\ & x \in K. \end{cases} \tag{7.7}$$

con $K = \{x \in \mathbb{R}^n : Ax = b, x \geqslant 0\}$, donde $A \in \mathbb{R}^{(m \times n)}$, y además rango$(A) = m < n$. Se tiene:

a. Si Inf $\{c^T x : x \in K\} \in \mathbb{R}$, entonces existe solución para el problema (PL).

b. Toda solución local es global.

Capítulo 8

Notación

Notación	**Descripción**
$\|x\|_0$	El número de coordenadas distintas de cero del vector x
$\|x\|_p$	$(\sum_{k=1}^{n} \|x_k\|^p)^{1/p}$
R_+	El intervalo $(0, +\infty)$
$\mathcal{A}(P)$	Conjunto de puntos admisibles del problema (P)
X	Dado $x \in \mathbb{R}^n$, X es la matriz diagonal tal que $X_{ii} = x_i$
$x \cdot s$	Dados $x, s \in \mathbb{R}^n$, $x \cdot s \in \mathbb{R}^n$ tal que $(x \cdot s)_i = x_i s_i$ para $1 \leqslant i \leqslant n$
máx$\{0, x\}$	Vector cuyas coordenadas son: máx$\{0, x_i\}$ para $1 \leqslant i \leqslant n$
$m \ll n$	m es mucho más pequeño que n
SVD	(Singular Value Decomposition) Descomposición en valores singulares

Bibliografía

[1] J. Nocedal y S. J. Wright, *Numerical Optimization*, Springer, New York, 2000.

[2] C. Moler, *"Magic" Reconstruction: Compressed Sensing*, http://es.mathworks.com/company/newsletters/articles/magic-reconstruction-compressed-sensing.html, 9/02/2016.

[3] A.M. Bruckstein, D.L. Donoho y M. Elad, *From Sparse Solutions of Systems of Equations to Sparse Modelling of Signal and Images*, SIAM Review, Vol. 51, Nº 1, páginas: 34-81, 2009.

[4] D. Mackenzie, *Compressed Sensing Makes Every Pixel Count*, Whats Happening in the Mathematical Sciences 7, American Math Society, 2009.

http://www.ams.org/samplings/math-history/hap7-pixel.pdf.

[5] E. Candes y J. Romberg, *l_1-MAGIC: Recovery of Sparse Signals via Convex Programming*, Caltech, Octubre de 2005.

[6] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[7] S. S. Chen, D. L. Donoho y M. A. Saunders, *Atomic Decomposition by Basis Pursuit*, SIAM Review, Vol. 43, No. 1, páginas: 129-159, 2001.

[8] *Discrete Cosine Transform*, https://en.wikipedia.org/wiki/Discrete_cosine_transform, Wikipedia.

[9] Y. Zhang *Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Enviroment*, Baltimore, Maryland, 1996.

[10] D. L. Donoho, *For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution*, Communications on Pure and Applied Mathematics, Vol. 59, Nº 6, páginas: 797-829, 2006.