

Treball final de màster
MÀSTER DE
MATEMÀTICA AVANÇADA
Facultat de Matemàtiques
Universitat de Barcelona

Mostreig de Compressió
Compressive Sampling

Sebastià Pons Burguera

Director:

F. Javier SORIA DE DIEGO

Realitzat al Departament de:

MATEMÀTICA APLICADA I ANÀLISI

Barcelona, 27 de juny de 2013

Índex

Agraïments	v
Resum	vii
Abstract	ix
1 Introducció	1
1.1 Motivació	1
1.2 Plantejament axiomàtic i objectius	2
1.3 Primeres aproximacions	3
1.3.1 Teorema fonamental de l'àlgebra lineal	3
1.3.2 Teorema de mostreig de Nyquist-Shannon	3
1.4 Mostreig de compressió	5
1.4.1 La compressibilitat és la clau	5
1.4.2 Minimització ℓ_1	6
1.4.3 Variants	7
1.5 Organització del treball	8
2 Preliminars	9

2.1	Operadors en espais de Hilbert	9
2.2	Transformada de Fourier	10
2.3	Wavelets i noiselets	11
2.4	Fórmula d'inclusió-exclusió	14
2.5	Nombres de Stirling	16
3	Mostreig de senyals	19
3.1	Detecció de la senyal	19
3.2	Dispersió i incoherència	19
3.2.1	Dispersió	20
3.2.2	Incoherència	21
3.3	Sistemes quasi ortonormals	22
4	El polinomi dual	27
4.1	Definició del polinomi	27
4.2	Moments de matrius aleatòries. Estimació clau	30
4.2.1	Primera fórmula per a l'esperança de la traça de $(H_0)^{2s}$. . .	30
4.2.2	Segona fórmula per a l'esperança de la traça de $(H_0)^{2s}$. . .	31
4.2.3	Estimació clau	33
4.3	Invertibilitat	36
4.4	Magnitud en T^c	38
5	Recuperació probable de senyals	45
5.1	Recuperació probablement exacta de senyals	45
5.2	Límit fonamental	51

5.3	Relació amb els principis d'incertesa	52
6	Recuperació determinista de senyals	55
6.1	Propietat de reconstrucció exacta	55
6.2	Propietats de reconstrucció dispersa dual	56
6.3	Reconstrucció determinista	60
6.4	Matrius aleatòries de Gauss	61
7	Exemples	65
7.1	ℓ_1 amb restriccions d'igualtat	65
7.1.1	Algorisme primal-dual	66
7.1.2	Concreció de l'algorisme	68
7.1.3	Rutines	70
7.1.4	Resultats	71
7.2	$TV(x)$ amb restriccions d'igualtat	76
7.2.1	Algorisme log-barrera	77
7.2.2	Concreció de l'algorisme	79
7.2.3	Rutines	81
7.2.4	Resultats	84
8	Conclusions	95
A	Codi font: exemple 7.1	99
A.1	Valors de recuperació mínima en $\ell_1(x)$	99
A.2	Recuperació mínima en $\ell_1(x)$	102
A.3	Solucionador de programació lineal (primal-dual)	105

A.4	Gradients conjugats	111
B	Codi font: exemple 7.2	115
B.1	Valors de recuperació en $TV(x)$ (color)	115
B.2	Valors de recuperació en $TV(x)$ (uniforme)	117
B.3	Valors de recuperació en $TV(x)$ (lineal)	119
B.4	Recuperació en $TV(x)$	120
B.5	Creació de la màscara uniforme	122
B.6	Creació de la màscara lineal	123
B.7	Constructor de la matriu de mesura	125
B.8	Constructor de matriu de mesura adjunta	125
B.9	Solucionador de programació lineal (log-barrera)	126
B.10	Newton per a $TV(x)$	129
B.11	Imatge aleatòria. Imitació “phantom”.	134
B.12	Representar el·lipse a base de píxels	135
B.13	Subprograma de “ellipseMatrix”	138
C	Figures de l'exemple 7.2	139
	Bibliografia	148

Agraïments

Gràcies al Màster de Matemàtica Avançada de la Universitat de Barcelona, una nova àrea de la matemàtica se m'obrí al davant. Amb assignatures de com la d'anàlisi harmònic, se'm despertà un interès per la matemàtica aplicada, concretament pel tractament d'imatges, del qual sorgeix aquest recull.

Agrair als companys del màster que durant aquest any m'han donat suport i m'han ajudat en el que he necessitat, així com als companys de pis i a la meva parella, per les hores que m'han acompanyat tot fent feina a l'hora de dur a terme aquest treball.

Finalment, agrair especialment al director del treball, Javier Soria, per acompanyar-me des de la distància en el desenvolupament del treball, donant-me idees, consell i suport per tal de dur a terme l'estudi que trobareu a continuació.

Resum

En moltes de les situacions de la vida real on hi intervé la captació o adquisició de senyals s'estableix una precisió, un màxim de mostres preses, que pot estar imposada voluntàriament o condicionada per mitjans tècnics. Pot ser que necessitem un temps que no tenim per mostrejar la senyal, imaginem el sensor d'una càmera fotogràfica; o que la informació és massa extensa per a captar-la tota, com és el cas de les tomografies.

Considerem una senyal $f(t) \in \mathbb{R}^n$ de la qual volem obtenir els n coeficients a partir de m mostres de la forma $y = \langle f, \varphi_k \rangle$, on $m \ll n$. El problema que se'ns planteja és el fet de minimitzar el nombre m de tal manera que puguem recuperar $f(t)$ d'una manera òptima. Ho farem minimitzant en norma ℓ_1 . La reconstrucció proposada serà $f^* = \Psi x^*$, on x^* , solució del problema d'optimització convexa

$$\min_{x \in \mathbb{R}^n} \|x\|_{\ell_1}, \quad \text{subjecte a} \quad y_k = \langle \varphi_k, \Psi x \rangle, \quad k = 1, \dots, m.$$

El *compressive sampling*, o mostreig de compressió, utilitza que la majoria de senyals que pertanyen al grup d'interès són de manera natural disperses o compressibles en el sentit que tenen representacions concises quan són expressades en una determinada base Ψ ; i el que anomenam incoherència, mitjançant un raonament dual expressa la idea de que els objectes que tenen una representació dispersa Ψ en un dels dominis n'han de tenir una d'extensa en l'altre.

Analitzant l'existència d'un polinomi dual –dependent de la mida del conjunt de mostres i del suport de la senyal original– i la probabilitat en que aquest satisfà dues propietats que anomenarem invertibilitat i magnitud en T^c , es fixen les hipòtesis i la probabilitat per a les quals la reconstrucció de la senyal és òptima. El resultat principal mostra que prenent una senyal de dimensió n , representada emprant Ψ de tal manera que la imatge sigui S -dispersa, mostrejada emprant Φ , de tal manera que s'obté un coeficient d'incoherència $\mu(\Phi, \Psi)$, llavors si prenem aleatòriament un nombre de mostres m obeint la fita

$$m \geq C_M \cdot \mu^2(\Psi, \Phi) \cdot S \cdot \log n,$$

llavors es pot reconstruir la senyal amb una probabilitat de $1 - O(n^{-M})$, on C_M és una constant que depèn del paràmetre de precisió M .

Seguint un raonament determinista, també mostrem com la senyal pot ser reconstruïda si $\delta_S + \theta_S + \theta_{S,2S} < 1$, on δ és la constant d'isometria restringida i θ la constant d'ortogonalitat restringida de la matriu de mostreig. Exemples de matrius que satisfan aquestes propietats són les matrius aleatòries de Gauss.

Amb tot això, tenim algorismes de presa de mostres i de recuperació de senyals factibles i, a més, que no depenen de la imatge en sí, sinó només de la mida de la senyal, del cardinal del suport de la senyal i de les bases de mostreig i representació. Per tant, els podem emprar per a un grans conjunts de senyals. En veim exemples en la darrera part del treball.

Abstract

In many real life situations where the signal acquisition or capture takes part in an accuracy is established. A maximum of samples, which may be voluntarily determined or imposed by technical means, are taken. Maybe you need a time that you don't have to sample the signal, imagine a camera sensor; maybe the information is too extensive to capture it all, as in the case of tomographies.

Let $f(t)$ be a signal in \mathbb{R}^n for which we want to get its n coefficients from m samples taken as $y = \langle f, \varphi_k \rangle$, where $m \ll n$. The dare now is how to minimize the samples m such that we can recover $f(t)$ in an optimal way. We will do that minimizing the ℓ_1 -norm. The proposed reconstruction is $f^* = \Psi x^*$, where x^* is solution of the convex optimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|_{\ell_1}, \quad \text{subject to} \quad y_k = \langle \varphi_k, \Psi x \rangle, \quad k = 1, \dots, m.$$

The *compressive sampling* uses that most of signals that belong to the set of interest are naturally sparse, or compressible, in the sense that they have concise representations when they are expressed in a particular basis Ψ ; and uses too a thing called incoherence, which through a dual reasoning, it expresses the idea that the objects which have a sparse representation Ψ in a domain have to have an extensive one in the other.

Analyzing the existence of a dual polynomial –which depends on the size of the sample set and the support of the original signal– and the probability in which two properties that we will call them invertibility and magnitude on T^c are satisfied, we set the assumptions and the probability for which the reconstruction of the signal is optimal. The main result shows that taking a signal of dimension n , represented using Ψ such that the image is S -sparse, sampled using Φ , and getting an incoherence coefficient $\mu(\Phi, \Psi)$, then if we take a random number of samples m obeying the bound

$$m \geq C_M \cdot \mu^2(\Psi, \Phi) \cdot S \cdot \log n,$$

the signal can be reconstructed with probability $1 - O(n^{-M})$, where C_M is a

constant value depending on the precision parameter M .

Following a deterministic argument, we show how the signal can be reconstructed if $\delta_S + \theta_S + \theta_{S,2S} < 1$ is fulfilled, where δ is the restricted isometry constant and θ the restricted orthogonality constant, both of the sampling matrix. Examples of matrices satisfying this properties are the Gauss random matrices.

With this, we have a feasible sampling and recovery algorithms for signals and, moreover, these don't depend on the image itself, these depend only on the size of the signal, the cardinal of the signal support and the sampling and representation bases. Because of that, we can use the same one for a big set of signals. We take a look at some examples in the last part of this work.

Capítol 1

Introducció

1.1 Motivació

En moltes de les situacions de la vida real on hi intervé la captació o adquisició de senyals s'estableix una precisió. Depenent del procés i de la senyal en si, la precisió pot estar imposada voluntàriament o condicionada per mitjans tècnics, d'emmagatzematge o de temps, entre altres factors. Per exemple, a l'hora de fer una fotografia, el sensor d'una càmera fotogràfica pren una certa informació de la imatge que es té al davant, però per tots és ben sabut que la qualitat de la imatge és limitada, i per tant, la càmera només guarda una part d'aquesta informació.

Per altra banda, en el cas de les ressonàncies magnètiques, la senyal a adquirir és una representació d'un tall transversal de l'individu. En aquest cas, s'entén la senyal com a un conjunt de freqüències les quals, canviades convenientment de base, es convertiran en la imatge desitjada. Aquest fet prové d'imaginar la llum com a ona, que generarà un color o un altre depenent de la freqüència d'aquesta.

En els exemples anteriors, el temps d'exposició per a la captació de la senyal és limitat i, per tant, el nombre de mostres que podem prendre també. En aquest context, és interessant trobar mètodes de mesura ràpids i, paral·lelament, dissenyar altres mètodes i algorismes que permetin recuperar la senyal a partir d'un mínim de mostres, de tal manera que puguem disminuir el nombre d'aquestes a prendre.

L'eficiència dels mitjans tècnics potser correspondrien més a un enginyer o a un informàtic, així que ens centrarem en la part de minimitzar el conjunt de mostres.

1.2 Plantejament axiomàtic i objectius

Considerem una senyal, que per ara podem imaginar com un vector. Es planteja la manera en que es podria recompondre la senyal original a partir d'una sèrie de mostres d'aquesta. La teoria del *compressive sampling*, o mostreig de compressió, desenvolupada per Candès i Tao [Can04], i de manera independent per Donoho [Don08], estudia sota quines condicions es pot recuperar l'esmentada senyal i fins a quin punt es pot minimitzar el conjunt de mostres per a obtenir la imatge original amb una qualitat òptima.

Sigui x una senyal tal que $x \in \mathbb{R}^n$. Prenguem un mostreig lineal y de dimensió m de la forma

$$y_i = \langle a_i, x \rangle, \quad i = 1, \dots, m. \quad (1.1)$$

Amb aquesta notació, hom pot escriure tal cosa com $y = Ax$, on A es una matriu de dimensió $m \times n$ amb els vectors a_i com a files.

Resultats com els que veurem a la secció 1.3 ens asseguren que podem reconstruir la senyal x a partir del mostreig y . En aquest treball anirem molt més enllà, estudiarem com hem d'agafar el mostreig per tal de “comprimir” les senyals a l'hora de prendre el mostreig i així reduir al màxim el nombre necessari de mostres per a la posterior recuperació. A més, ho farem des d'un punt de vista general en quant a les senyals, això és, que el procés volem que sigui independent del contingut de senyal. Un altre fet important és que no sols ens quedarem en l'existència teòrica de la recuperació, sinó que es pretén que la senyal pugui ser realment reconstruïda.

Per tant, els objectius que es plantegen es podrien resumir en els següents punts.

- Trobar bases de mostreig de senyals que permetin la compressió d'aquestes en el sentit que es pugui minimitzar el mostreig per a la posterior recuperació.
- Fixar una fita per al valor m de mostres necessàries per a la recuperació real de senyals, així com estudiar si existeix algun límit fonamental per a tal valor.
- Obtenir algorismes de recuperació, ja sigui amb arguments probabilístics o deterministes, però independents de la senyal i de tal manera que es pugui aplicar de manera generalista.
- Per mitjà d'exemples pràctics, assegurar que els resultats anteriors siguin vàlids i que es pugui dur a terme la reconstrucció de diversos tipus de senyals.

1.3 Primeres aproximacions

Des de molts d'àmbits de la matemàtica, hi ha molts resultats que asseguruen la resolució de fórmules del tipus $y = Ax$, on en el nostre cas y pren la forma que trobem en (1.1).

1.3.1 Teorema fonamental de l'àlgebra lineal

La primera idea que ens ve al cap és utilitzar el Teorema fonamental de l'àlgebra lineal. Una de les maneres de formular-lo és la següent.

Teorema 1.3.1. *Sigui $A \in \mathbb{R}^{m \times n}$, llavors,*

- $\dim R(A) = \dim R(A^T)$
- $\dim N(A) = n - r$
- $\dim N(A^T) = m - r$

on r és el rang de la matriu, $N(A)$ el seu nucli, $R(A)$ el seu espai columna i $R(A^T)$ el seu co-nucli.

Segons el que ens diu el teorema, necessitem m equacions per a trobar les n incògnites, amb la qual cosa se'n dedueix que ha de ser $m = n$.

Evidentment aquesta solució queda descartada, ja que el mostreig ha de ser la senyal sencera.

1.3.2 Teorema de mostreig de Nyquist-Shannon

Si canviem de branca i anem a l'anàlisi harmònic trobem el teorema de mostreig de Shannon [Her96].

Teorema 1.3.2. *Suposem $f \in L^1(\mathbb{R})$ i $\text{supp}(\hat{f}) \subseteq [-B\pi, B\pi]$. Llavors,*

$$f(x) = \sum_{k \in \mathbb{Z}} f\left(\frac{k}{B}\right) \frac{\sin \pi(Bx - k)}{\pi(Bx - k)},$$

on les sumes parcials simètriques d'aquesta sèrie convergeixen a f en norma L^2 .

Observem que aquest teorema afirma que per a una funció de banda limitada en $L^1(\mathbb{R})$ tal que $\text{supp}(\hat{f}) \subseteq [-B\pi, B\pi]$ està unívocament determinada per un mostreig de f als punts $\{B^{-1}k : k \in \mathbb{Z}\}$ i la reconstrucció ve donada per la família de funcions

$$\left\{ \frac{\sin \pi(Bx - k)}{\pi(Bx - k)} : k \in \mathbb{Z} \right\}.$$

família relacionada amb la funció $\text{sinc}(x)$, que és una funció d'escalat de la *wavelet* de Shannon.

Un teorema estretament relacionat és la versió de Shannon del Teorema de Nyquist:

Teorema 1.3.3. *Si una funció $x(t)$ no conté freqüències més grans que B hertz, aquesta està completament determinada donant les ordenades a sèries de punts espaiats per un màxim de $1/(2B)$ segons.*

En altres paraules, una funció de banda limitada pot ser reconstruïda perfectament per una seqüència finita de mostres si l'ample de banda B no és major que la meitat de la taxa de mostreig (mostres per segon), o el que és el mateix, si la freqüència de mostreig és major que el doble de la freqüència màxima de la senyal.

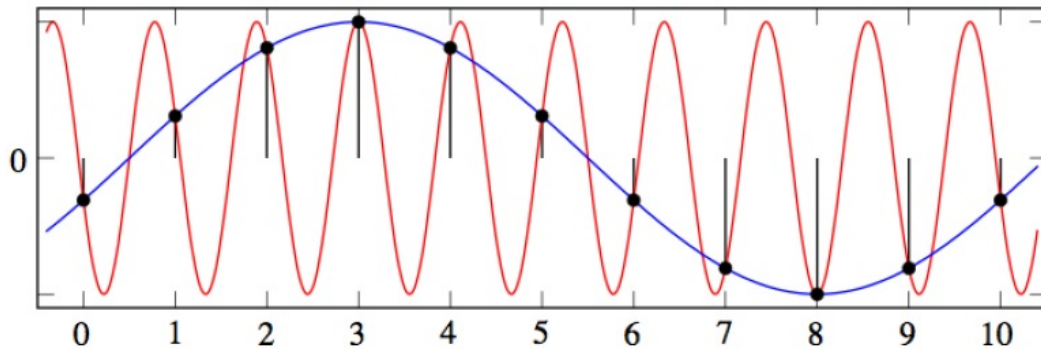


Figura 1.1: Mostreig de 10 punts, menor al doble de la freqüència de mostreig

D'aquesta manera, obtenim una relació de la forma $m < n$, on la diferència depèn de la senyal concreta. En cas contrari pot passar el que observem a la figura 1.1.

1.4 Mostreig de compressió

La teoria desenvolupada per Candès i Tao afirma que es pot reconstruir una senyal a partir d'un mostreig molt petit de la imatge, $m \ll n$. Aquest mostreig és aleatori en algunes versions i, en altres, donat per una màscara de mostreig que no dependrà mai de la senyal, sinó del mètode.

L'algorisme utilitzat per recuperar la imatge es basa en resoldre de manera convenient el següent programa d'optimització.

$$\min \|x\|_{\ell_1} \quad \text{subjecte a} \quad Ax = y.$$

1.4.1 La compressibilitat és la clau

La clau en el *compressive sampling* o mostreig de compressió és la compressibilitat de la senyal [Can08]. Hom ja sap que podem comprimir matrius disperses i matrius quasi disperses a través d'algorismes com l'RLE, basat en un recorregut per la senyal codificant-la per parells on en la segona posició hi ha un valor distint a zero i en la primera el nombre de zeros que s'han trobat en l'esmentat recorregut des del darrer valor distint a zero. Aquest tipus de codificació sovint va precedida d'algorismes que imposen un llindar de forma controlada, de tal manera que forcen zeros en determinades zones de la senyal sense una pèrdua excessiva en la qualitat d'aquesta. Un exemple seria l'algorisme JPEG.

En el nostre cas, l'algorisme de Candès i Tao aconsegueix reconstruir la senyal acuradament si aquesta és aproximadament dispersa i pràcticament exacta quan la senyal és dispersa. Aquest fet és sorprenent donat que la manera en que prenem les mostres està fixada prèviament, sense dependre de la senyal.

Aquest fet es fonamenta en dur a terme el mostreig en una base determinada de tal forma que, amb un canvi de base adequat, puguem representar-lo d'una manera, com hem dit, dispersa.

Així, amb el *compressive sampling* ens n'adonem que hom pot dissenyar un algorisme de mostreig no adaptat de tal manera que condensa la informació d'una senyal compressible en una quantitat reduïda de dades. El nombre de mostres necessari serà comparable al nivell de dispersió de la senyal i no a la grandària d'aquesta ni a la situació del seu suport.

Aquí hi ha dos principis treballant. El primer és que el mostreig aleatori, o quasi aleatori, pot ser emprat com un mecanisme efectiu per a comprendre senyals a partir d'ones incoherents, quasi aleatòries, extraient-ne la informació de manera eficient a partir de poques mostres. L'altre és el que la minimització ℓ_1 és capaç de desembullar les components de la senyal x a partir de les dades comprimides.

1.4.2 Minimització ℓ_1

Però quin rol juga la norma ℓ_1 en les matrius disperses?

Trobar solucions disperses a sistemes d'equacions indeterminats és, en general, un problema *NP*-difícil. Per exemple, suposem $y = Ax = Bx + e$, i definim la matriu F de tal manera que $FB = 0$. Llavors, la solució més dispersa ve donada per

$$\min_{d \in \mathbb{R}^n} \|d\|_{\ell_0}, \quad \text{subjecte a } Fd = \tilde{y} (= Fe), \quad (1.2)$$

i sense altre coneixement, resoldre aquest problema requereix essencialment cerques exhaustives sobre tots els subconjunts de columnes de F . Aquest procediment, clarament combinatori, té una complexitat exponencial.

Són molts els treballs que han duit a terme recerca per trobar alternatives a (1.2). Per exemple, a [Can05],

$$\min_{d \in \mathbb{R}^n} \|d\|_{\ell_1}, \quad \text{subjecte a } Fd = \tilde{y}, \quad (1.3)$$

on recordem que $\|d\|_{\ell_1} = \sum_{i=1}^m |d_i|$. Així com la norma ℓ_0 detecta el nombre de coordenades distintes a zero, la norma ℓ_1 és convexa, i les solucions d'ambdós problemes són equivalents [Can04b]. A més, la solució del programa (1.3) pot ser trobada mitjançant programació lineal.

Així, veim que la norma ℓ_1 és un funcional que “promou”, ajuda a trobar, senyals donades per matrius disperses [Can08]. Resultats amb la recuperació amb la norma ℓ_1 no són nous, ja foren estudiats a principis dels anys 60. Suposem

$$y(t) = f(t) + n(t), \quad t \in \mathbb{R},$$

on $f(t)$ té un ample de banda limitat, $f(t) \in B(\Omega) := \{f : \hat{f}(\omega) = 0 \text{ for } |\omega| > \Omega\}$, i $n(t)$ és un soroll a impulsos amb suport en un conjunt dispers T . S'ha demostrat el següent fenomen: si recuperam f amb

$$\min_{\tilde{f} \in B(\Omega)} \|y - \tilde{f}\|_{\ell_1(\mathbb{R})},$$

llavors la recuperació és exacta mentre $|T||\Omega| \leq \pi/2$. Això passa per a qualsevol $f \in B(\Omega)$ i per a qualsevol grandària del soroll. El fet principal és que f és dispersa en el domini de freqüències i, sota aquestes hipòtesis, la minimització ℓ_1 pot donar lloc a resultats inesperats.

1.4.3 Variants

L'algorisme presentat té diverses variants [Can04] que responen a diferents situacions com és, per exemple, senyals molt grans. Alguns exemples són els següents:

- Min- ℓ_1 amb igualtat com a restricció:

$$\min \|x\|_1 \quad \text{subjecte a} \quad Ax = b.$$

- Error d'aproximació ℓ_1 mínim:

$$\min_x \|y - Ax\|_1.$$

- Min- ℓ_1 amb restricció quadràtica:

$$\min \|x\|_1 \quad \text{subjecte a} \quad \|Ax - b\|_2 \leq \varepsilon.$$

- Min- ℓ_1 amb correlació residual fitada o seleccionador *Dantzig*

$$\min \|x\|_1 \quad \text{subjecte a} \quad \|A^*(Ax - b)\|_\infty \leq \gamma.$$

Donada una senyal en dues dimensions, una imatge, un model alternatiu és que el gradient sigui dispers. Si x_{ij} és el píxel de la fila i columna j d'una imatge de $n \times n$, definim l'operador

$$D_{ij}x = \begin{pmatrix} D_{h;ij}x \\ D_{v;ij}x \end{pmatrix},$$

on

$$D_{h;ij}x = \begin{cases} x_{i+1,j} - x_{ij} & \text{si } i < n \\ 0 & \text{si } i = n \end{cases} \quad \text{i} \quad D_{v;ij}x = \begin{cases} x_{i,j+1} - x_{ij} & \text{si } j < n \\ 0 & \text{si } j = n \end{cases}.$$

Llavors, $D_{ij}x$ pot ser interpretat com un tipus de gradient discret. En aquest cas, la variació total ve definida com segueix.

$$TV(x) := \sum_{ij} \sqrt{(D_{h;ij}x)^2 + (D_{v;ij}x)^2} = \sum_{ij} \|D_{ij}x\|_2. \quad (1.4)$$

D'aquesta manera, podem considerar les següents variants per a dues dimensions.

- Min- TV amb igualtat com a restricció:

$$\min TV(x) \quad \text{subjecte a} \quad Ax = b.$$

- Min- TV amb restricció quadràtica:

$$\min TV(x) \quad \text{subjecte a} \quad \|Ax - b\|_2 \leq \varepsilon.$$

- *Dantzig TV*

$$\min TV(x) \quad \text{subjecte a} \quad \|A^*(Ax - b)\|_\infty \leq \gamma.$$

Estudiarem principalment la primera de les varietats esmentades, tot i que implementarem la minimització de $TV(x)$ amb igualtat com a restricció per tal de comprovar l'exactitud de la reconstrucció per al cas bidimensional.

1.5 Organització del treball

Aquest treball està dividit en 4 blocs. El primer està format per la secció 2, on hi ha definicions i resultats previs que emprarem en els altres blocs. En el pròxim bloc, la secció 3, s'analitzen les propietats que ha de tenir el mostreig i quina influència té sobre el nombre de mostres necessari per a la reconstrucció de la senyal. El tercer bloc està format per les seccions 4, 5 i 7. En la primera part d'aquest bloc s'analitza l'existència d'un polinomi dual –depenent de la mida del conjunt de mostres i del suport de la senyal original–, i la probabilitat en que aquest satisfà dues propietats que anomenarem invertibilitat i magnitud en T^c . Utilitzant aquest fet, es demostra en la segona secció d'aquest bloc la possibilitat de reconstrucció de la senyal tot fixant una fita en el nombre de mostres necessari. En el final d'aquest bloc, que es troba ja al final del treball, hi ha exemples pràctics de recuperació en probabilitat, tot provant els resultats demostrats sobre distints tipus de senyals. El darrer bloc està format per la secció 6, on es demostra sota quines condicions la recuperació determinista és possible, així com també es mostra que les matrius aleatòries de Gauss les satisfan amb una probabilitat molt alta.

Capítol 2

Preliminars

En aquest capítol trobareu definicions i alguns resultats que s'empren en algun moment del treball, així com alguns conceptes que anirà bé tenir en compte per tal de copsar bé la teoria desenvolupada.

2.1 Operadors en espais de Hilbert

Definició 2.1.1. Sigui H un espai de Hilbert tal que el seu producte interior ve donat per $(\cdot, \cdot)_H = (\cdot, \cdot)$. Per a tot $x, y \in H$ és compleix

$$|(x, y)| \leq \|x\| \cdot \|y\|,$$

on $\|z\| = (z, z)^{1/2}$, i s'anomena *desigualtat de Cauchy-Schwartz*.

Definició 2.1.2. Siguin H_1, H_2 dos espais de Hilbert i sigui

$$T : H_1 \rightarrow H_2$$

una aplicació lineal i continua. Es defineix la seva norma com

$$\|T\| = \sup_{\|x\|_{H_1} \leq 1} \|Tx\|_{H_2} = \sup_{\substack{\|x\|_{H_1} \leq 1 \\ \|y\|_{H_2} \leq 1}} |(Tx, y)_{H_2}|.$$

Definició 2.1.3. L'*operador adjunt* de $T : H_1 \rightarrow H_2$ és un altre operador

$$T^* : H_2 \rightarrow H_1$$

tal que

$$(Tx, y)_{H_2} = (x, y^*)_{H_1}, \quad \forall x \in H_1.$$

Definició 2.1.4. $T : H \rightarrow H$ és *autoadjunt* si $T = T^*$.

Definició 2.1.5. Donat $\{w_k\}_{k \in \mathbb{N}}$, direm que és un *sistema ortonormal* si $\|w_k\| = 1$ i $(w_j, w_k) = 0 \ \forall k \neq j$.

Proposició 2.1.6. $\mathbb{B} = \{w_j\}_{j=1}^n \subset H$ és una *base ortonormal* si, i només si, $W = \{w_{j,k}\}_{j,k=1}^n$ és una *matriu unitària*:

$$W \cdot \bar{W}^t = I_n, \quad \text{i.e.,} \quad W^{-1} = \bar{W}^t,$$

on I_n és la *matriu identitat de dimensió n* .

2.2 Transformada de Fourier

Transformada discreta i transformada ràpida de Fourier

Proposició 2.2.1. El sistema discret d'exponencials trigonomètriques $\{E_{m,n}\}_{m,n}$, definit per

$$E_{m,n}(j, k) = \frac{1}{\sqrt{MN}} e^{-2\pi i \left(\frac{(j-1)(m-1)}{M} + \frac{(k-1)(n-1)}{N} \right)},$$

és *ortonormal*.

Definició 2.2.2. Donada una matriu $A \in \ell^2(\mathbb{Z}/M \times \mathbb{Z}/N)$, es defineix la seva *transformada de Fourier discreta (DFT)* com la matriu \hat{A} , de la mateixa dimensió que A , on per a cada $1 \leq m \leq M$ i $1 \leq n \leq N$,

$$\hat{A}(m, n) = \frac{1}{\sqrt{MN}} \sum_{j=1}^M \sum_{k=1}^N A(j, k) e^{-2\pi i \left(\frac{(j-1)(m-1)}{M} + \frac{(k-1)(n-1)}{N} \right)}.$$

I la seva inversa com

$$\check{A}(m, n) = \frac{1}{\sqrt{MN}} \sum_{j=1}^M \sum_{k=1}^N A(j, k) e^{2\pi i \left(\frac{(j-1)(m-1)}{M} + \frac{(k-1)(n-1)}{N} \right)},$$

de tal manera que $\check{\check{A}} = A$.

Observació 2.2.3. Per a vectors, la mateixa fórmula és vàlida posant $M = 1$. En aquest cas, l'algorisme DFT necessita N^2 productes. Existeix una variant anomenada *transformada ràpida de Fourier (FFT)* que disminueix l'ordre fins a $N \log N$ productes.

Definició 2.2.4. Un sistema ortogonal $\{w_k\}_{k \in \mathbb{N}}$ és *complet* si es compleix que, si $\hat{x}(k) = 0 \forall k$, llavors $x = 0$.

Definició 2.2.5. Direm que un vector està *localitzat* prop de j_0 si quasi tots les seves components són zero tret de j_0 i valors petits al seu entorn.

Observació 2.2.6. La base estàndard $e_k = (\overbrace{0, \dots, 1, \dots}^k, 0)$ és la que està més ben localitzada. Per altra banda, la transformada n -dimensional de Fourier, corresponent a la base d'exponencials complexes és la que està pitjor localitzada, cada posició de W , donada per $w_{j,k} = \frac{1}{\sqrt{n}} e^{-2\pi i \left(\frac{(j-1)(k-1)}{n} \right)}$ té norma $1/\sqrt{n}$.

Identitat de Parseval

Definició 2.2.7. Donada B una base ortonormal, es defineix la *identitat de Parseval* com

$$\|x\|^2 = \langle x, x \rangle = \sum_{v \in B} |\langle x, v \rangle|^2.$$

Propietat 2.2.8. Es compleix que $\|A\|_{\ell^2(\mathbb{Z}/M \times \mathbb{Z}/N)} = \|\hat{A}\|_{\ell^2(\mathbb{Z}/M \times \mathbb{Z}/N)}$.

Teorema 2.2.9 (Riesz-Fischer). *Un sistema ortogonal $\{w_k\}_{k \in \mathbb{N}}$ és complet si, i només si, per a tot x en un espai de Hilbert H ,*

$$x = \sum_{k=1}^{\infty} \hat{x}(k) w_k,$$

i es compleix la identitat de Parseval

$$\|x\|^2 = \sum_{k=1}^{\infty} |\hat{x}(k)|^2.$$

2.3 Wavelets i noiselets

Wavelets

La base estàndard està ben localitzada en el domini de l'espai, però no en el de freqüències, al contrari que passa amb la transformada de Fourier. Les *wavelets* [Fra99] són un tipus de bases ortonormals que tenen molt bones propietats de localització, tant en el domini espacial com el de freqüències.

Definició 2.3.1. Definim el funcional $R_k w$ com la translació $R_k w(j) = w(j - k)$.

En [Fra99] es demostra que $\{R_k w\}$ és una base ortonormal si, i només si $|\hat{w}(j)| = 1/\sqrt{n}$ per a tot j .

Definició 2.3.2. Considerem N un enter parell ($N = 2M$, per algun $M \in \mathbb{N}$). Una base ortonormal per a $\ell^2(\mathbb{Z}/N)$ de la forma $\{R_{2k} u\}_{k=0}^{M-1} \cup \{R_{2k} v\}_{k=0}^{M-1}$, per algun $u, v \in \ell^2(\mathbb{Z}/N)$, s'anomena la base de *wavelet* de primera etapa per a $\ell^2(\mathbb{Z}/N)$.

Exemples

Definició 2.3.3 (Transformada de Haar). Considerem un vector x de dimensió N , es defineixen els coeficients $a^1 = (\langle R_0 u, x \rangle, \dots, \langle R_{2(M-1)} u, x \rangle)$ anomenat tendència, i $d^1 = (\langle R_0 v, x \rangle, \dots, \langle R_{2(M-1)} v, x \rangle)$, anomenat detalls. Es defineix, per $N = 2^k$ la *transformada de Haar d'ordre j* com

$$\text{Haar}_j(x) = (\text{Haar}_{j-1}(a^{j-1}), d^{j-1}, \dots, d^1) = (a^j, d^j, d^{j-1}, \dots, d^1),$$

per a $j = 2, \dots, k$.

Definició 2.3.4 (Base real de Shannon). Donat un enter N divisible per 4, es defineix aquesta *wavelet*, ben localitzada en el domini de freqüències, mitjançant els vectors

$$\hat{u}(j) = \begin{cases} \sqrt{\frac{2}{N}} & \text{si } j = 1, \dots, N/4 - 1 \text{ o } j = 3N/4 + 1, \dots, N, \\ \frac{i}{\sqrt{N}} & \text{si } j = N/4, \\ \frac{-i}{\sqrt{N}} & \text{si } j = 3N/4, \\ 0 & \text{si } j = 1, \dots, N/4 + 1 \text{ o } j = 3N/4 - 1, \dots, N, \end{cases}$$

i

$$\hat{v}(j) = \begin{cases} 0 & \text{si } j = 1, \dots, N/4 - 1 \text{ o } j = 3N/4 + 1, \dots, N, \\ \sqrt{\frac{1}{N}} & \text{si } j = N/4 \text{ o } j = 3N/4, \\ \sqrt{\frac{2}{N}} & \text{si } j = N/4 + 1, \dots, 3N/4 - 1. \end{cases}$$

Definició 2.3.5 (wavelet D6 de Daubechies). Donat un enter N parell i no menor a 8, es defineix aquesta *wavelet*, molt ben localitzada en l'espai, segons els vectors $u, v \in \ell^2(\mathbb{Z}/N)$:

$$u = \frac{\sqrt{2}}{32} \left[2a + 3b + 3c, 6a + 4b + 2c, 6a + 4b - 2c, 2a + 3b - 3c, b - c, \overbrace{0, \dots, 0}^{N-6}, b + c \right]$$

i

$$v = \left[u(N), \overbrace{0, \dots, 0}^{N-6}, -u(5), u(4), -u(3), u(2), -u(1) \right],$$

on $a = 1 - \sqrt{10}$, $b = 1 + \sqrt{10}$ i $c = \sqrt{5 + 2\sqrt{10}}$.

Noiselets

El terme noiselet es refereix a una família de funcions que estan relacionades amb les wavelets de la mateixa manera en què la base de Fourier es relaciona amb una senyal en el domini de temps. En altres paraules, que si una senyal és compacta en el domini de wavelets, aquesta és extensa en el domini de noiselets i viceversa.

La complementarietat de wavelets i noiselets ens dona l'oportunitat d'emprar noiselets en mostreig de compressió per reconstruir una senyal que té una representació compacta en el domini de wavelets.

Foren introduïdes per primera vegada en el treball de Coifman, Geshwind i Meyer [Coi01]. Són funcions semblants al soroll, d'aquí el seu nom, en el sentit que són totalment incomprensibles per qualsevol mètode basat en wavelets ortogonals. D'aquesta manera, no es poden transformar de tal manera que no semblin soroll emprant cap tipus de mètode que empli la dualitat temps-freqüència.

Definició 2.3.6. L'anàlisi de multiresolució de Haar en $[0, 1]$ està definit per

$$V_n = \{f \in L^2([0, 1]) \mid f \text{ constant per tot interval de la forma } (k2^{-n}, (k+1)2^{-n})\}.$$

Definició 2.3.7. Per a tot x en $[0, 1]$, considerem la família de funcions definides de manera recursiva

$$\begin{aligned} f_1(x) &= \chi_{[0,1]}(x), \\ f_{2n}(x) &= (1 - i)f_n(2x) + (1 + i)f_n(2x - 1), \\ f_{2n+1}(x) &= (1 + i)f_n(2x) + (1 - i)f_n(2x - 1). \end{aligned}$$

Observació 2.3.8. La família $\{f_j\}$ de funcions té tots els seus coeficients de Haar-Walsh de mòdul 1, fins en el millor escalat possible. Les distribucions limitadores tenen totes uns coeficients de Haar-Walsh ben definits, tots de mòdul 1. A més, les funcions tenen suport en $[0, 1]$, on tenen valor absolut constant.

Teorema 2.3.9. El conjunt $\{f_j \mid j = 2^N, \dots, 2^{N+1} - 1\}$ és una base ortogonal de V_N .

2.4 Fórmula d'inclusió-exclusió

Considerem un conjunt A i $\mathcal{P}(A)$ el conjunt de relacions d'equivalència en A . Observem que hi ha un ordenament parcial en les relacions d'equivalència, ja que hom pot dir que $\sim_1 \leq \sim_2$ si \sim_1 és més gruixuda que \sim_2 , és a dir, si $a \sim_2 b$ implica que $a \sim_1 b$ per a tot $a, b \in A$. D'aquesta manera, l'element més rude o gruixut en $\mathcal{P}(A)$ és la relació en la qual tots els elements de A són equivalents (només una classe d'equivalència). Mentre que l'element més fi és la relació d'igualtat $=$, és a dir, que cada element pertany a una classe d'equivalència distinta ($|A|$ classes d'equivalència).

Considerem ara el conjunt G de cardinal $\mathcal{P}(A)$. Per cada relació d'equivalència \sim de \mathcal{P} definim els conjunts $\Omega(\sim) \subset G$ i $\Omega_{\leq}(\sim) \subset G$ com

$$\begin{aligned}\Omega(\sim) &:= \{\omega \in G : \sim_{\omega} = \sim\}, \\ \Omega_{\leq}(\sim) &:= \bigcup_{\sim' \in \mathcal{P} : \sim' \leq \sim} \Omega(\sim') = \{\omega \in G : \sim_{\omega} \leq \sim\}.\end{aligned}$$

Els conjunts $\{\Omega_{\leq}(\sim) : \sim \in \mathcal{P}\}$ formen una partició de G . Aquests conjunts $\Omega_{\leq}(\sim)$ també es poden definir com

$$\Omega_{\leq}(\sim) := \{\omega \in G : \omega_a = \omega_b \text{ sempre que } a \sim b\}.$$

De manera similar, els conjunts $\Omega(\sim)$ es poden definir com

$$\Omega(\sim) := \{\omega \in G : \omega_a = \omega_b \text{ sempre que } a \sim b \text{ i } \omega_a \neq \omega_b \text{ sempre que } a \not\sim b\}.$$

Exemple 2.4.1. Suposem $A = \{1, 2, 3, 4\}$ i $G = \mathbb{Z}_n^4$ per a un cert n natural positiu. Fixem \sim tal que $1 \sim 4$ i $2 \sim 3$ (exactament dues classes d'equivalència). Llavors, $\Omega(\sim) := \{\omega \in \mathbb{Z}_n^4 : \omega_1 = \omega_4, \omega_2 = \omega_3 \text{ i } \omega_1 \neq \omega_2\}$, mentre que $\Omega_{\leq}(\sim) := \{\omega \in \mathbb{Z}_n^4 : \omega_1 = \omega_4 \text{ i } \omega_2 = \omega_3\}$.

Lema 2.4.2 (Principi d'inclusió-exclusió per a classes d'equivalència). *Siguin A i G conjunts no buits i finits. Per a cada classe d'equivalència $\sim \in \mathcal{P}(A)$ on $\omega \in G^{|A|}$, es té que*

$$\sum_{\omega \in \Omega(\sim)} f(\omega) = \sum_{\sim_1 \in \mathcal{P} : \sim_1 \leq \sim} (-1)^{|A/\sim| - |A/\sim_1|} \left(\prod_{A' \in A/\sim_1} (|A'/\sim| - 1)! \right) \sum_{\omega \in \Omega_{\leq}(\sim_1)} f(\omega), \quad (2.1)$$

on A/\sim denota les classes d'equivalència de \sim . D'aquesta manera, si prenem $A = \{1, 2, 3\}$ i \sim ve donada per $j \sim k$ si, i només si, $j = k$, llavors la identitat

anterior afirma que

$$\begin{aligned} & \sum_{\omega_1, \omega_2, \omega_3 \in G: \omega_1, \omega_2, \omega_3 \text{ distints}} f(\omega_1, \omega_2, \omega_3) = \sum_{\omega_1, \omega_2, \omega_3 \in G} f(\omega_1, \omega_2, \omega_3) \\ & - \sum_{\omega_1, \omega_2, \omega_3 \in G: \omega_1 = \omega_2} f(\omega_1, \omega_2, \omega_3) - \sum_{\omega_1, \omega_2, \omega_3 \in G: \omega_2 = \omega_3} f(\omega_1, \omega_2, \omega_3) \\ & - \sum_{\omega_1, \omega_2, \omega_3 \in G: \omega_3 = \omega_1} f(\omega_1, \omega_2, \omega_3) + 2 \sum_{\omega_1, \omega_2, \omega_3 \in G: \omega_1 = \omega_2 = \omega_3} f(\omega_1, \omega_2, \omega_3). \end{aligned}$$

Demostració. Passant A a l'espai quocient A/\sim si és necessari, hem d'assumir que la relació \sim és $=$. Ara, canviant les etiquetes d' A per $\{1, 2, \dots, n\}$, \sim per \sim_1 i A' per A , el que hem de provar és

$$\begin{aligned} & \sum_{\omega \in \Omega^n: \omega_1, \dots, \omega_n \text{ distints}} f(\omega) = \\ & \sum_{\sim_1 \in \mathcal{P}(\{1, \dots, n\})} (-1)^{n-|\{1, \dots, n\}/\sim_1|} \left(\prod_{A \in \{1, \dots, n\}/\sim_1} (|A| - 1)! \right) \sum_{\omega \in \Omega_{\leq}(\sim_1)} f(\omega). \end{aligned} \quad (2.2)$$

Ho provarem per inducció sobre n . Quan $n = 1$ ambdues parts de la igualtat són iguals a $\sum_{\omega \in G} f(\omega)$. Ara suposem que és cert per $n - 1$ i provem-ho per n .

Observem que la part esquerra de l'equació (2.2) es pot escriure com

$$\sum_{\omega' \in G^{n-1}: \omega_1, \dots, \omega_{n-1} \text{ distints}} \left(\sum_{\omega_n \in G} f(\omega', \omega_n) - \sum_{j=1}^{n-1} f(\omega', \omega_j) \right),$$

on $\omega' := (\omega_1, \dots, \omega_{n-1})$. Aplicant-hi la hipòtesi d'inducció, això s'escriu com

$$\begin{aligned} & \sum_{\sim' \in \mathcal{P}(\{1, \dots, n-1\})} (-1)^{n-1-|\{1, \dots, n-1\}/\sim'|} \prod_{A' \in \{1, \dots, n-1\}/\sim'} (|A'| - 1)! \\ & \times \sum_{\omega' \in \Omega_{\leq}(\sim')} \left(\sum_{\omega_n \in G} f(\omega', \omega_n) - \sum_{j=1}^n f(\omega', \omega_j) \right). \end{aligned} \quad (2.3)$$

Ara fixem-nos en la part dreta de (2.2). Si \sim és una classe d'equivalència en $\{1, \dots, n\}$, sigui \sim' la restricció de \sim a $\{1, \dots, n-1\}$. Noti's que \sim pot ser format a partir de \sim' adjuntant el singletó $\{n\}$ com una nova classe d'equivalència (en aquest cas, escrivim $\sim = \{\sim', \{n\}\}$), o triant $j \in \{1, \dots, n-1\}$ i declarant n equivalent a j (en aquest cas, escrivim $\sim = \{\sim', \{n\}\}/(j = n)$). Notem que, amb la

darrera construcció, podem recuperar la mateixa classe d'equivalència \sim de moltes maneres si la classe d'equivalència $[j]_{\sim'}$ de j en \sim' és més gran que 1, llavors ho podem resoldre assignant un pes a cada j de $\frac{1}{|[j]_{\sim'}|}$. Llavors, tenim la identitat

$$\begin{aligned} \sum_{\sim \in \mathcal{P}(\{1, \dots, n\})} F(\sim) &= \sum_{\sim' \in \mathcal{P}(\{1, \dots, n-1\})} F(\{\sim', \{n\}\}) \\ &+ \sum_{\sim' \in \mathcal{P}(\{1, \dots, n-1\})} \sum_{j=1}^{n-1} \frac{1}{|[j]_{\sim'}|} F(\{\sim', \{n\}\} / (j=n)), \end{aligned}$$

per a qualsevol funció F a valors complexos sobre $\mathcal{P}(\{1, \dots, n\})$. Aplicant això sobre la part dreta de (2.2), ho podem reescriure com a la suma de

$$\sum_{\sim' \in \mathcal{P}(\{1, \dots, n-1\})} (-1)^{n-(|\{1, \dots, n-1\}/\sim'|+1)} \left(\prod_{A \in \{1, \dots, n-1\}/\sim'} (|A| - 1)! \right) \sum_{\omega' \in \Omega_{\leq}(\sim')} f(\omega', \omega_n)$$

i

$$\sum_{\sim' \in \mathcal{P}(\{1, \dots, n-1\})} (-1)^{n-|\{1, \dots, n-1\}/\sim'|} \sum_{j=1}^{n-1} T(j) \sum_{\omega' \in \Omega_{\leq}(\sim')} f(\omega', \omega_n),$$

on adoptem la convenció $\omega' = (\omega_1, \dots, \omega_{n-1})$. Com que

$$T(j) := \frac{1}{|[j]_{\sim'}|} \prod_{A \in \{1, \dots, n\}/(\{\sim', \{n\}\}/(j=n))} (|A| - 1)! = \prod_{A' \in \{1, \dots, n-1\}/\sim'} (|A'| - 1)!,$$

llavors la part dreta de (2.2) coincideix amb (2.3) conclouent la demostració. \square

2.5 Nombres de Stirling

Definició 2.5.1. Per a cada $n, k \geq 0$ definim el *nombre de Stirling de segon tipus*, $S(n, k)$, com el nombre de relacions d'equivalència en un conjunt de n elements que té exactament k classes d'equivalència. Ho podem escriure com

$$S(n, k) = \#\{\sim \in \mathcal{P}(A) : |A/\sim| = k\}.$$

Propietat 2.5.2. Els nombres de Stirling compleixen la recurrència

$$S(n+1, k) = S(n, k+1) + kS(n, k) \text{ per a tot } k, n \geq 0. \quad (2.4)$$

Aquesta propietat ens diu que, si a és un element de A i \sim és una relació d'equivalència en A amb k classes d'equivalència, o bé a no és equivalent a cap dels altres elements d' A (i per tant tendríem que \sim té $k - 1$ classes d'equivalència en $A \setminus \{a\}$), o bé a és equivalent a un de les k classes d'equivalència de $A \setminus \{a\}$.

Propietat 2.5.3. Els nombres de Stirling compleixen la identitat

$$S(n+1, k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} (k-i)^n.$$

Aquesta propietat prové d'aplicar inducció a la Propietat 2.5.2.

Lema 2.5.4. Per a cada $n \geq 1$ i $0 \leq \tau \leq 1/2$, es compleix la identitat

$$\sum_{k=1}^n (k-1)! S(n, k) (-1)^{n-k} \tau^k = \sum_{k=1}^{\infty} (-1)^{n-k} \frac{\tau^k k^{n-1}}{(1-\tau)^k}. \quad (2.5)$$

Demostració. Ho provarem per inducció sobre n . Quan $n = 1$, la banda esquerra és igual a τ , i la banda dreta és igual a

$$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\tau^k}{(1-\tau)^k} = - \sum_{k=0}^{\infty} \left(\frac{\tau}{1-\tau} \right)^k + 1 = \frac{-1}{1 - \frac{\tau}{1-\tau}} + 1 = \tau,$$

com volíem. Ara suposem-ho cert per $n-1$ i provem-ho per n . Aplicant l'operador $(\tau^2 - \tau) \frac{d}{d\tau}$ a ambdós costats, cosa que ve justificada a partir de la hipòtesi $0 \leq \tau \leq 1/2$, i després d'algunes simplificacions obtenim

$$\sum_{k=1}^{n+1} (k-1)! (S(n, k-1) + kS(n, k)) (-1)^{n+1-k} \tau^k = \sum_{k=0}^{\infty} (-1)^{n+1-k} \frac{\tau^k k^n}{(1-\tau)^k},$$

d'on es té el resultat aplicant-hi (2.4). □

Per a facilitar la notació, definirem $F_n(\tau)$ com a la quantitat (2.5), és a dir,

$$F_n(\tau) = \sum_{k=1}^n (k-1)! S(n, k) (-1)^{n-k} \tau^k = \sum_{k=1}^{\infty} (-1)^{n-k} \frac{\tau^k k^{n-1}}{(1-\tau)^k}. \quad (2.6)$$

D'aquesta manera, tenim $F_1(\tau) = \tau$, $F_2(\tau) = -\tau + \tau^2$, $F_3(\tau) = \tau - 3\tau^2 + 2\tau^3$, etcètera. Notem també que, per valors molt petits de τ , tenim l'aproximació $|F_n(\tau)| \approx (-1)^{n+1} \tau$. Altres fites més estrictes les presentam a continuació.

Lema 2.5.5. *Sigui $n \geq 1$ i $0 \leq \tau \leq 1/2$. Si $\frac{\tau}{1-\tau} \leq e^{1-n}$, llavors tenim que $|F_n(\tau)| \leq \frac{\tau}{1-\tau}$. Si, a més, $\frac{\tau}{1-\tau} > e^{1-n}$, llavors*

$$|F_n(\tau)| \leq \exp((n-1) \left(\log(n-1) - \log \log \frac{1-\tau}{\tau} - 1 \right)).$$

Demostració. Fent càlculs elementals mostra que, per $x > 0$, la funció $g(x) = \frac{\tau^x x^{n-1}}{(1-\tau)^x}$ és creixent per $x < x_*$ i decreixent per $x > x_*$, on $x_* := (n-1)/\log \frac{1-\tau}{\tau}$. Si $\frac{\tau}{1-\tau} \leq e^{1-n}$, llavors $x_* \leq 1$, i llavors la sèrie alternada $F_n(\tau) = \sum_{k=1}^{\infty} (-1)^{n+k} g(k)$ té magnitud com a màxim $g(1) = \frac{\tau}{1-\tau}$. D'altra banda, la sèrie té magnitud com a màxim

$$g(x_*) = \exp((n-1) \left(\log(n-1) - \log \log \frac{1-\tau}{\tau} - 1 \right)),$$

d'on obtenim el resultat. □

Amb això veim que $F_n(\tau)$ es comporta com τ , per n de l'ordre de $O(\log[1/\tau])$, i com $(n/\log[1/\tau])^n$, per n molt més gran de $\log[1/\tau]$.

Ens serà d'utilitat representar la fita trobada com a

$$F_n(\tau) \leq G(n),$$

on

$$G(n) = \begin{cases} \frac{\tau}{1-\tau}, & \log \frac{\tau}{1-\tau} \leq 1-n \\ \exp((n-1)(\log(n-1) - \log \log \frac{\tau}{1-\tau} - 1)), & \log \frac{\tau}{1-\tau} > 1-n. \end{cases} \quad (2.7)$$

Noti's que hem intercanviat els papers dels arguments de la funció per reflectir la idea de que hem de veure G com a una funció de n amb paràmetre τ .

Capítol 3

Mostreig de senyals

3.1 Detecció de la senyal

Considerem una senyal $f(t) \in \mathbb{R}^n$. Prenguem un mostreig lineal de dimensió m a partir de funcionals de la forma

$$y_k = \langle f, \varphi_k \rangle, \quad k = 1, \dots, m. \quad (3.1)$$

La idea principal és reconstruir la senyal $f(t)$ a partir del mostreig y . Els funcionals intenten aproximar cada y_k per una forma d'ona φ_k . Per exemple, si les formes d'ona són deltes de Dirac, és a dir, pics, llavors el vector y és un vector de mostres de f en el domini de temps o d'espai. Si són funcions característiques de píxels, llavors y és la imatge presa pel sensor d'una càmera fotogràfica digital. En el cas en que les formes d'ona corresponguin a funcions sinusoidals, y correspondrà a un vector de coeficients de Fourier, seria el que obtendríem en les ressonàncies magnètiques.

3.2 Dispersió i incoherència

El mostreig de compressió es basa en dos conceptes clau, la dispersió, que té a veure amb les senyals d'interès, i la incoherència, que té a veure amb la modalitat de detecció.

A través del concepte de dispersió, el mostreig de compressió explota el fet que

determinades senyals que pertanyen al grup d'interès són de manera natural disperses o compressibles en el sentit que tenen representacions concises quan són expressades en una determinada base Ψ .

Per altra banda, la incoherència estén la dualitat existent entre el domini de temps i de freqüències i expressa la idea de que els objectes que tenen una representació dispersa Ψ en un dels dominis n'han de tenir una d'extensa en l'altre.

El fet important aquí és que hom pot dissenyar protocols de detecció o mostreig que capturin la informació rellevant incrustada en la senyal dispersa i condensar-la en un petit conjunt de dades. Aquests protocols són no adaptats i sols requereixen la correlació de la senyal amb un nombre fix de formes d'ona que són incoherents amb la base oferint una descripció concisa de la senyal.

3.2.1 Dispersió

Moltes de les senyals habituals tenen una representació concisa quan són expressades en una base convenientment triada. Si ens imaginem qualsevol imatge aleatòria en gama de grisos, llavors quasi cap píxel tindrà valor zero, la informació estarà molt dispersa. Tot i així, en el domini de freqüències, la informació de la imatge es concentra en pocs coeficients rellevants, sent la majoria d'ones moltes petites.

Matemàticament parlant, considerem in vector $f \in \mathbb{R}^n$, on n és el nombre de píxels de la imatge considerada. Sigui $\Psi = [\psi_1 \psi_2 \dots \psi_n]$ una base ortonormal de wavelets en la qual estenem la imatge de la següent manera:

$$f(t) = \sum_{i=1}^n x_i \psi_i(t), \quad (3.2)$$

on x correspon als coeficients de la seqüència de f , $x_i = \langle f, \psi_i \rangle$. Serà convenient expressar f com a Ψx on Ψ és la matriu de $n \times n$ amb les ψ_i com a columnes.

Definició 3.2.1. Un vector (o matriu) és *dispers* si la majoria dels seus coeficients són zero. Si el nombre de coeficients distints a zero és S , direm que és *S-dispers*.

D'aquesta manera, queda clar la implicació que té la dispersió i, per tant, podem obviar la majoria de coeficients ja que no tenen un paper fonamental en la representació de la imatge. Considerem x_S el vector de coeficients x quedant-nos sols amb els S majors i fent zero els altres. Llavors, definim $f_S := \Psi x_S$, que és dispers, de fet, *S-dispers*.

Observació 3.2.2. Notem que hem pres la base Ψ ortonormal, i per tant tenim

$$\|f - f_S\|_{\ell_2} = \|x - x_S\|_{\ell_2},$$

i si x és dispers o compressible, en el sentit que les magnituds triades dels coeficients en x decauen ràpidament, llavors x és aproximadament x_S , ja que l'error $\|f - f_S\|_{\ell_2}$ és petit.

En aquest punt cal notar que el coneixement sobre l'espai de freqüències de la senyal és necessari per tal de quedar-nos amb els coeficients més importants i per tant comprimir-la òptimament. Tot i així, aquest treball pretén anar més lluny, ja que l'objectiu és treballar amb un mètode no adaptat, és a dir, que no depèn de la senyal. Per a realitzar aquesta tasca la dispersió de la qual parlem no té un paper insignificant, és una eina fonamental de la modelització que ens permetrà un processat de la senyal eficient. En resum del que veurem, la dispersió de la imatge ens dirà amb quina eficiència podrem adquirir senyals de manera no adaptada.

3.2.2 Incoherència

Suposem un parell (Φ, Ψ) de bases ortonormals de \mathbb{R}^n . La primera base Φ l'empram per al procés de detecció de la senyal f i la segona per a representar f com en (3.1). El fet que les bases siguin ortonormals no és imprescindible, les prenem així per tal de facilitar la notació i els càlculs. Tot i així, pot ser interessant que ho siguin per tenir propietats tan bones com la de l'observació 3.2.2.

Definició 3.2.3. La *coherència* entre el sistema de detecció Φ i el sistema de representació Ψ ve donat per

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k, l \leq n} |\langle \varphi_k, \psi_l \rangle|.$$

Per tant, la coherència mesura la màxima correlació entre cada dos elements de Φ i Ψ . Si les bases contenen elements correlacionats la coherència és alta, si no, és baixa.

Observació 3.2.4. Es compleix que $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$. L'extrem superior s'obté de si tots els productes $|\langle \varphi_k, \psi_l \rangle|$ entre parells de vectors unitaris és 1. L'extrem inferior és una conseqüència de la identitat de Parseval, la qual afirma que per a cada j , $\sum_{k=1}^n |\langle \varphi_k, \psi_j \rangle|^2 = \|\psi_j\|_{\ell_2}^2 = 1$.

El mostreig de compressió té a veure principalment amb parells de bases entre les quals hi ha una baixa coherència. A continuació hi ha uns quants exemples d'aquests tipus de parells.

Exemple 3.2.5. Prenem Φ la base canònica o la base de pics definida a partir de $\varphi_k(t) = \delta(t - k)$ i Ψ la base de Fourier definida per $\psi_j(t) = n^{-1/2}e^{2\pi i jt/n}$. Com que la matriu Ψ és la de detecció, correspondrà a l'esquema de mostreig clàssic en el temps o en l'espai. Els parells temps-freqüència compleixen $\mu(\Phi, \Psi) = 1$, i per tant tenen una incoherència maximal. El fet presentat està pensat en una dimensió, però aquests pics i sinusoidals són incoherents en qualsevol dimensió, ja que el resultat s'estén per a imatges en dues dimensions, en tres dimensions, etcètera.

Exemple 3.2.6. En aquest cas, prenem Ψ una base de wavelets i Φ una base de noiselets. La coherència entre noiselets i wavelets de Haar és de $\sqrt{2}$, entre els noiselets i les wavelets de Daubechies D4 i D8 és aproximadament 2.2 i 2.9, respectivament, si prenem un ampli conjunt de mostres de tamany n . Noti's que aquest fet val per dimensió 1 i s'estén a dimensions superiors. L'interès en les noiselets prové del fet que (3.1) són incoherents amb els sistemes que proveeixen informació dispersa de senyals, i que per a (3.2) existeixen algorismes molt eficients d'ordre $O(n)$, i així com passa per la transformada de Fourier, no es necessita emmagatzemar la matriu de noiselets per aplicar-la a un vector. Aquest fet és molt important per als algorismes que utilitzats per al mostreig de compressió, ja que sense ells la computació resultaria massa poc pràctica.

3.3 Sistemes quasi ortonormals

En aquesta secció veurem algunes propietats sobre els conjunts del mostreig relacionat amb la base en la qual ho representem.

Com hem dit amb anterioritat, en principi no és necessari utilitzar bases ortonormals per a l'estudi de les propietats que fins ara hem exposat. Però respecte la recuperació de senyals, aquest fet tornarà de gran importància per temes d'unicitat.

Notació 3.3.1. Denotarem per $(v_j)_{j \in J} \in \mathbb{R}^p$ les columnes d'una matriu F i per H l'espai de Hilbert generat per aquests vectors. A més, per cada $T \subseteq J$, denotarem per F_T a la submatriu amb columnes d'índexs $j \in T$, llavors

$$F_T c = \sum_{j \in T} c_j v_j \in H.$$

Per introduir el concepte de sistema quasi ortonormal, primer fem la següent observació. Si les columnes de F són massa “degenerades”, llavors el problema de la reconstrucció no pot ser resolt. En particular, si existeix una combinació lineal dispersa però no trivial $\sum_{j \in T} c_j v_j = 0$ de v_j tal que sumen zero, i si $T = T_1 \cup T_2$ tal que $T_1 \cap T_2 = \emptyset$, llavors el vector y definit per

$$y := \sum_{j \in T_1} c_j v_j = \sum_{j \in T_2} (-c_j) v_j$$

té dues representacions disperses. Per altra banda, dependències lineals de la forma $\sum_{j \in J} c_j v_j = 0$, les quals impliquen un gran nombre de coeficients c_j distints de zero, com a oposat a l'existència d'un conjunt dispers de coeficients, no presenta una obstrucció evident a la reconstrucció de la senyal dispersa. L'altre extrem seria el que presentarem, si $(v_j)_{j \in J}$ formen un sistema ortonormal, llavors el problema de la reconstrucció és “fàcilment” resolt definint $c_j = \langle f, v_j \rangle_H$.

Una manera de suavitzar la hipòtesis de tenir uns sistemes ortonormals de detecció o representació de la senyal és prendre el que denominarem *hipòtesis d'ortonormalitat restringida*. Amb aquesta hipòtesi, que és molt més feble que el fet de que els sistemes implicats siguin ortonormals, podem resoldre el problema de la recuperació tot i que $(v_j)_{j \in J}$ siguin altament linealment dependents. Per exemple, és possible per a $m := |J|$ molt més gran que la dimensió de l'espai general per les v_j .

Definició 3.3.2. (Constants d'isometria restringida) Sigui F la matriu amb la col·lecció finita de vectors $(v_j)_{j \in J} \in \mathbb{R}^p$ com a columnes. Per a cada enter $1 \leq S \leq |J|$, definim les *constants d'isometria S -restringida* δ_S com el nombre més petit tal que F_T compleix

$$(1 - \delta_S) \|c\|^2 \leq \|F_T c\|^2 \leq (1 + \delta_S) \|c\|^2, \quad (3.3)$$

per a tots els subconjunts $T \subset J$ de cardinal menor o igual a S i tots els coeficients reals $(c_j)_{j \in T}$. De la mateixa manera, definim les *constants d'ortogonalitat S, S' -restringida* $\theta_{S, S'}$ per a $S + S' \leq |J|$ com el nombre més petit tal que es compleix

$$|\langle F_T c, F_{T'} c' \rangle| \leq \theta_{S, S'} \|c\| \|c'\|, \quad (3.4)$$

per a tots els subconjunts $T, T' \subset J$ disjunts i de cardinals tal que $|T| \leq S$ i $|T'| \leq S'$.

Els nombres δ_S i θ_S mesuren com d'aprop estan els vectors v_j de comportar-se com un sistema ortonormal, però sols restringint l'atenció a les combinacions lineals

disperses que incloguin no més de S vectors. Aquests nombres són no decreixents en S, S' . Per $S = 1$, el valor δ_1 només transmet informació sobre els vectors v_j ; de fet, δ_1 és la millor constant tal que

$$1 - \delta_1 \leq \|v_j\|_H^2 \leq 1 + \delta_1, \quad \text{per a tot } j \in J.$$

En particular, es té que $\delta_1 = 0$ si, i només si, tots els v_j tenen mòdul 1.

Definició 3.3.3. Direm que la matriu de mostreig F compleix el *principi d'incertesa uniforme* si compleix (3.3).

El següent lema ens mostra que com major són els δ_S més control tenim sobre la ortogonalitat dels $\theta_{S,S'}$. Aquest, ens dona control sobre l'amplitud de l'angle principal entre els subespais de dimensió S i S' respectivament. Serà útil perquè ens ajudarà a garantir la reconstrucció exacta a partir de sols el coneixement dels δ .

Lema 3.3.4. *Es compleix $\theta_{S,S'} \leq \delta_{S+S'} \leq \theta_{S,S'} + \max(\delta_S, \delta_{S'})$, per a tot S, S' .*

Demostració. Primer provarem que $\theta_{S,S'} \leq \delta_{S+S'}$: Per homogeneïtat serà suficient provar que

$$\left| \left\langle \sum_{j \in T} c_j v_j, \sum_{j' \in T'} c'_{j'} v_{j'} \right\rangle_H \right| \leq \delta_{S+S'},$$

sempre que $|T| \leq S$, $|T'| \leq S'$, T i T' disjunts i $\sum_{j \in T} |c_j|^2 = \sum_{j' \in T'} |c'_{j'}|^2 = 1$. Llavors, a partir de (3.3) tenim

$$2(1 - \delta_{S+S'}) \leq \left\| \sum_{j \in T} c_j v_j + \sum_{j' \in T'} c'_{j'} v_{j'} \right\|_H^2 \leq 2(1 + \delta_{S+S'})$$

i també

$$2(1 - \delta_{S+S'}) \leq \left\| \sum_{j \in T} c_j v_j - \sum_{j' \in T'} c'_{j'} v_{j'} \right\|_H^2 \leq 2(1 + \delta_{S+S'}),$$

i el resultat es dedueix aplicant la llei del paral·lelogram

$$\langle f, g \rangle = \frac{\|f + g\|_H^2 - \|f - g\|_H^2}{4}.$$

Resta provar que $\delta_{S+S'} \leq \theta_{S,S'} + \max(\delta_S, \delta'_S)$.

Un altre cop per homogeneïtat, serà suficient veure que

$$\left| \left\langle \sum_{j \in \tilde{T}} c_j v_j, \sum_{j' \in \tilde{T}} c_{j'} v_{j'} \right\rangle_H - 1 \right| \leq \theta_S + \max(\delta_S, \delta'_S)$$

sempre que $|\tilde{T}| \leq S + S'$ i $\sum_{j \in \tilde{T}} |c_j|^2 = 1$. Per provar-ho, fem una partició de \tilde{T} tal que $\tilde{T} = T \cup T'$, on $|T| \leq S$ i $|T'| \leq S'$, i escriguem $\sum_{j \in T} |c_j|^2 = \alpha$ i $\sum_{j \in T'} |c_j|^2 = 1 - \alpha$. Juntant (3.3) i (3.4) obtenim

$$\begin{aligned} (1 - \delta_S)\alpha &\leq \left\langle \sum_{j \in T} c_j v_j, \sum_{j' \in T} c_{j'} v_{j'} \right\rangle_H && \leq (1 + \delta_S)\alpha, \\ (1 - \delta_{S'})(1 - \alpha) &\leq \left\langle \sum_{j \in T'} c_j v_j, \sum_{j' \in T'} c_{j'} v_{j'} \right\rangle_H && \leq (1 + \delta_{S'})(1 - \alpha), \\ & \left| \left\langle \sum_{j \in T} c_j v_j, \sum_{j' \in T} c_{j'} v_{j'} \right\rangle_H \right| && \leq \theta_{S,S'} \alpha^{1/2} (1 - \alpha)^{1/2}. \end{aligned}$$

Llavors,

$$\begin{aligned} \left| \left\langle \sum_{j \in T} c_j v_j, \sum_{j' \in T} c_{j'} v_{j'} \right\rangle_H \right| &\leq \delta_S \alpha + \delta_{S'} (1 - \alpha) + 2\theta_{S,S'} \alpha^{1/2} (1 - \alpha)^{1/2} \\ &\leq \theta_{S,S'} + \max(\delta_S, \delta'_S). \end{aligned}$$

□

La rellevància dels nombres δ_S d'isometria respecte el problema de recuperació de senyals disperses és profunda. El següent lema ens dóna la unicitat de solució amb la que abans observàvem problemes.

Lema 3.3.5. *Suposem que $S \geq 1$ i tal que $\delta_{2S} < 1$, i també considerem $T \subset J$ tal que $|T| \leq S$. Sigui $f := F_T c$ per algun vector aleatori c de dimensió $|T|$. Llavors, el conjunt T i els coeficients $(c_j)_{j \in T}$ poden ser reconstruïts de manera única a partir del coneixement del vector f i dels coeficients v_j .*

Demostració. Provarem que hi ha un sol c tal que $\|c\|_{\ell_0} \leq S$ i complint $f = \sum_j c_j v_j$. Suposem, per contradicció, que f té dues representacions disperses $f = F_T c = F_{T'} c'$, on $|T|, |T'| \leq S$. Llavors, definint

$$d_j := c_j 1_{j \in T} - c'_j 1_{j \in T'},$$

tenim que

$$F_{T \cup T'} d = 0.$$

Si prenem normes a cada banda i hi aplicam (3.3) i la hipòtesis $\delta_{2S} < 1$ tenim que $\|d\|^2 = 0$, contradient la hipòtesis que deia que les dues representacions eren distintes. \square

Notem que el lema previ només ens dóna l'existència d'una solució teòricament possible, de cap manera un mètode ni algorisme per a la recuperació de T i dels c_j a partir de f i $(v_j)_{j \in J}$. En el capítol 6 veurem resultats per a la recuperació que garantiran que trobem el resultat desitjat per mitjà d'un algorisme conegut.

Capítol 4

El polinomi dual

4.1 Definició del polinomi

Considerem un vector $f \in \mathbb{C}^n$ amb suport $\text{supp } f = T$. Definim el vector signe com $\text{sgn}(f)(t) = f(t)/|f(t)|$ quan $t \in T$ i $\text{sgn}(f)(t) = 0$ en altre cas. Si definim $\ell_2(T)$ (respectivament $\ell_2(\Omega)$) com l'espai de senyals que són zero fora de T (respectivament Ω), podem definir la transformada de Fourier restringida $\mathcal{F}_{T \rightarrow \Omega} : \ell_2(T) \rightarrow \ell_2(\Omega)$ com a

$$\mathcal{F}_{T \rightarrow \Omega} f := \hat{f}|_{\Omega} \quad \text{per a tot } f \in \ell_2(T).$$

Representarem per $\mathcal{F}_{\Omega} = \mathcal{F}_{\mathbb{Z}_n \rightarrow \Omega}$ la transformada de Fourier del vector f restringint després a Ω ; l'operador $\iota : \ell_2(T) \rightarrow \ell_2(\mathbb{Z}_n)$ estén un vector de T a un vector en \mathbb{Z}_n posant zeros fora de T ; i $\iota^* f = f|_T$ és l'aplicació de restricció dual.

Llavors, definim el polinomi trigonomètric P com:

$$P := \mathcal{F}_{\Omega}^* \mathcal{F}_{T \rightarrow \Omega} (\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega})^{-1} \iota^* \text{sgn } f. \quad (4.1)$$

És fàcil veure que \hat{P} té suport en Ω i que

$$P(t) = \text{sgn}(f)(t), \quad \text{per a tot } t \in T,$$

ja que $\iota^* \mathcal{F}_{\Omega}^* = \mathcal{F}_{T \rightarrow \Omega}^*$, i per tant $\iota^* P = \iota^* \text{sgn}(f)$.

El que demostrarem en el que resta de secció és que també compleix les dues propietats següents:

1. *Invertibilitat.* L'operador $\mathcal{F}_{T \rightarrow \Omega}$ és injectiu. Per tant, el terme $\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}$ de (4.1) és invertible amb probabilitat $1 - O(n^{-M})$.
2. *Magnitud en T^c .* La funció P en (4.1) satisfà $|P(t)| < 1$ per a tot $t \in T^c$ amb probabilitat $1 - O(n^{-M})$.

Observació 4.1.1. Un detall que ens pot cridar l'atenció de la propietat 1 és que un funcional sigui invertible amb una certa probabilitat. Per entendre-ho, recordem que el funcional de $\mathcal{F}_{T \rightarrow \Omega}$, directament relacionat amb la transformada de Fourier, pot ser expressat en forma de matriu. Llavors, la invertibilitat d'aquest vendrà determinada per la invertibilitat de la matriu. Però existeix una mesura que digui quant de lluny està una matriu de ser singular? Doncs sí. Així com el valor propi major és una aproximació de la norma de la matriu, llavors el menor és una estimació de la distància a la que està la matriu al conjunt de matrius singulars. Per tant, hem de pensar que una matriu és invertible amb una certa probabilitat si hi ha aquesta probabilitat de que el menor valor propi sigui igual a zero i, per tant, la matriu sigui singular. Una primera aproximació dels resultats que derivarem d'aquest capítol directament relacionada amb els valors propis de matrius binomials, de Gauss i de Fourier, és a [Can04b].

El model Bernoulli

Treballar directament en el cas on Ω d'una determinada mida és triat aleatòriament amb una distribució uniformement pot ser molt complicat, ja que la probabilitat que una certa freqüència estigui inclosa o no en Ω depèn d'on estan o no incloses la resta de freqüències. Per això, introduïrem el següent model de probabilitat per a la selecció de Ω , que després traslladarem al model uniforme desitjat.

Definició 4.1.2. Un conjunt Ω' de coeficients de Fourier és mostrejat segons un *model Bernoulli* amb paràmetre $0 < \tau < 1$ creant la seqüència

$$I_\omega = \begin{cases} 0 & \text{amb probabilitat } 1 - \tau, \\ 1 & \text{amb probabilitat } \tau \end{cases}, \quad (4.2)$$

i llavors definim Ω' com

$$\Omega' := \{\omega : I_\omega = I\}. \quad (4.3)$$

Noti's que la mida del conjunt Ω' , $|\Omega'|$, és també aleatòria, seguint una distribució binomial, i $\mathbb{E}(|\Omega'|) = \tau N$. De fet, emprant arguments probabilístics clàssics es sap que, per n gran, $|\Omega'|/n \approx \tau$ amb una probabilitat alta.

Reescrivint P mitjançant operadors

Des d'ara en endavant suposarem $|\tau n| > M \log n$. En el capítol 5 s'explicarà el perquè d'aquesta suposició.

Ens serà d'ajuda reescriure (4.1) en termes de la matriu auxiliar

$$Hf(t) := - \sum_{\omega \in \Omega} \sum_{t' \in T: t' \neq t} e^{2\pi i \frac{\omega(t-t')}{n}} f(t') \quad (4.4)$$

i definir

$$H_0 = \iota^* H.$$

Aquest fet prové de les següents relacions:

$$\begin{aligned} \iota - \frac{1}{|\Omega|} H &= \frac{1}{|\Omega|} \mathcal{F}_\Omega^* \mathcal{F}_{T \rightarrow \Omega}, \\ I_T - \frac{1}{|\Omega|} H_0 &= \frac{1}{|\Omega|} \mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}, \end{aligned}$$

on I_T és la identitat per $\ell^2(T)$. Per deduir la primera relació és suficient substituir les funcions de l'esquerra i agrupar convenientment per veure com s'igualen al terme de l'esquerra. Noti's que $\iota^* \iota = I_T$ i que $\iota^* \mathcal{F}_\Omega^* = \mathcal{F}_{T \rightarrow \Omega}^*$; afegint això a la definició de H_0 , obtenim la segona relació a partir d'aplicar ι^* a la primera.

D'aquesta manera,

$$P = \left(\iota - \frac{1}{|\Omega|} H \right) \left(I_T - \frac{1}{|\Omega|} H_0 \right)^{-1} \iota^* \operatorname{sgn} f.$$

El detall aquí està en separar la diagonal constant de $\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}$ ($= |\Omega| I_T - H_0$), que és $|\Omega|$ per tot, del que està fora de la diagonal, que oscil·la. Triant Ω aleatòriament, podem veure H_0 com una matriu de "soroll", fent que $I_T - \frac{1}{|\Omega|} H_0$ estigui ben condicionat.

4.2 Moments de matrius aleatòries. Estimació clau

En aquesta secció demostrarem el que serà una estimació clau en les dues darres seccions d'aquest mateix capítol. En la subsecció 4.2.1 donarem una primera expansió de l'expressió $\mathbb{E}(\text{Tr}(H_0^{2s}))$. Com que no serà fàcil de manipular, necessitarem reescriure-la utilitzant les Fórmula d'inclusió-exclusió presentada en la secció 2.4 i les Identitats dels nombres de Stirling, presentades en la secció 2.5. Tot això ho farem en la subsecció 4.2.2, establint una segona expansió de $\mathbb{E}(\text{Tr}(H_0^{2s}))$ molt més fàcil de controlar. La prova del teorema passarà per fitar els termes individuals de la segona expansió, ho farem en la subsecció 4.2.3.

4.2.1 Primera fórmula per a l'esperança de la traça de $(H_0)^{2s}$

Recordem que $H_0(t, t')$, $t, t' \in T$, és la matriu de $|T| \times |T|$ amb entrades definides per

$$H_0(t, t') = \begin{cases} 0, & t = t' \\ c(t - t'), & t \neq t', \end{cases} \quad c(u) = \sum_{\omega \in \Omega} e^{\frac{2\pi i}{n} \omega u}.$$

Un element de la diagonal de la potència parell $2s$ de H_0 s'expressa com

$$H_0^{2s} = \sum_{t_2, \dots, t_{2s}: t_j \neq t_{j+1}} c(t_1 - t_2) \dots c(t_{2s} - t_1),$$

adoptant la convenció de que $t_{2s+1} = t_1$ on pertoqui i, per tant,

$$\mathbb{E}(\text{Tr}(H_0^{2s})) = \sum_{t_2, \dots, t_{2s}: t_j \neq t_{j+1}} \mathbb{E} \left[\sum_{\omega_1, \dots, \omega_{2s} \in \Omega} e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})} \right].$$

Emprant (4.3) i la linealitat de l'esperança ho podem escriure com

$$\mathbb{E}(\text{Tr}(H_0^{2s})) = \sum_{t_1, \dots, t_{2s}: t_j \neq t_{j+1}} \sum_{0 \leq \omega_1, \dots, \omega_{2s} \leq n-1} e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})} \mathbb{E} \left[\prod_{j=1}^{2s} I_{\{\omega_j \in \Omega\}} \right]. \quad (4.5)$$

La idea aquí és emprar la independència dels $I_{\omega_j \in \Omega}$ per a simplificar l'expressió. Per fer-ho, haurem d'anar alerta, ja que alguns ω_j poden ser el mateix, i per tant hom perd, en aquest punt, la independència de les variables indicador.

Per tal d'atacar aquest problema anem a definir la notació que emprarem.

Diguem $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ el conjunt de seqüències com hem emprat ja abans, i sigui A el conjunt finit $A := \{1, \dots, 2s\}$. Per a tot $\boldsymbol{\omega} := (\omega_1, \dots, \omega_{2s})$ definim la relació d'equivalència $\sim_{\boldsymbol{\omega}}$ en A com $j \sim_{\boldsymbol{\omega}} j'$ si, i només si, $\omega_j = \omega_{j'}$. Diguem $\mathcal{P}(A)$ al conjunt de totes les relacions d'equivalència en A .

Per cada una de les relacions d'equivalència \sim de \mathcal{P} definim els conjunts $\Omega(\sim) \subset \mathbb{Z}_n^{2s}$ i $\Omega_{\leq}(\sim) \subset \mathbb{Z}_n^{2s}$ com en la secció 2.4, per a l' A definit ara i $G = \mathbb{Z}_n^{2s}$.

Com que les variables aleatòries I_k definides en (4.2) són independents i tenen la mateixa distribució, la quantitat $\mathbb{E}[\prod_{j=1}^{2s} I_{\omega_j}]$ depèn només de la relació d'equivalència $\sim_{\boldsymbol{\omega}}$ i no dels valors concrets de $\boldsymbol{\omega}$. Per tant, tenim

$$\mathbb{E}[\prod_{j=1}^{2s} I_{\omega_j}] = \tau^{|A/\sim|},$$

on A/\sim denota les classes d'equivalència de \sim . Per tant, podem reescriure l'equació (4.5) com

$$\mathbb{E}(\text{Tr}(H_0^{2s})) = \sum_{t_1, \dots, t_{2s}: t_j \neq t_{j+1}} \sum_{\sim \in \mathcal{P}(A)} \tau^{|A/\sim|} \sum_{\boldsymbol{\omega} \in \Omega(\sim)} e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})}, \quad (4.6)$$

on \sim recorre totes les relacions d'equivalència.

4.2.2 Segona fórmula per a l'esperança de la traça de $(H_0)^{2s}$

La secció 2.4, referida a la Fórmula d'inclusió-exclusió ens ajuda a reescriure sumes sobre $\Omega(\sim)$ en termes de sumes sobre $\Omega_{\leq}(\sim)$. Si recordem i comparem les definicions d'aquests dos conjunts, entenem que aconseguim eliminar de l'expressió els termes no coincidents.

La suma interior de (4.6) es pot escriure com

$$\sum_{\sim \in \mathcal{P}(A)} \tau^{|A/\sim|} \sum_{\boldsymbol{\omega} \in \Omega(\sim)} f(\boldsymbol{\omega}),$$

on $f(\boldsymbol{\omega}) := e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})}$.

Lema 4.2.1.

$$\sum_{\sim \in \mathcal{P}(A)} \tau^{|A/\sim|} \sum_{\boldsymbol{\omega} \in \Omega(\sim)} f(\boldsymbol{\omega}) = \sum_{\sim_1 \in \mathcal{P}(A)} \left[\sum_{\boldsymbol{\omega} \in \Omega_{\leq}(\sim_1)} f(\boldsymbol{\omega}) \right] \prod_{A' \in A/\sim_1} F_{|A'|}(\tau). \quad (4.7)$$

Demostració. Aplicant (2.1) i reajustant, ho podem escriure com

$$\sum_{\sim_1 \in \mathcal{P}(A)} T(\sim_1) \sum_{\omega \in \Omega_{\leq}(\sim_1)} f(\omega),$$

on

$$T(\sim_1) = \sum_{\sim \in \mathcal{P}(A): \sim \geq \sim_1} \tau^{|A/\sim|} (-1)^{|A/\sim| - |A/\sim_1|} \prod_{A' \in A/\sim_1} (|A'/\sim| - 1)!$$

Si separam A en les classes d'equivalència A' de A/\sim_1 , tenim

$$T(\sim_1) = \prod_{A' \in A/\sim_1} \sum_{\sim' \in \mathcal{P}(A')} \tau^{|A'/\sim'|} (-1)^{|A'/\sim'| - |A'|} (|A'/\sim'| - 1)!$$

Si ara separam les \sim' segons el nombre de classes d'equivalència $|A'/\sim'|$, podem escriure

$$T(\sim_1) = \prod_{A' \in A/\sim_1} \sum_{k=1}^{|A'|} |A'| S(|A'|, k) \tau^k (-1)^{|A'| - k} (k - 1)! = \prod_{A' \in A/\sim_1} F_{|A'|}(\tau),$$

gràcies al Lema 2.5.4 sobre els nombres Stirling i a la notació presentada en (2.6). Tornant a reescriure $T(\sim_1)$, com en la darrera equació, al seu lloc en la fórmula obtenim el resultat. \square

Si prenem l'equació (4.7) per a la funció $f(\omega) := e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})}$, obtenim

$$\mathbb{E}[\text{Tr}(H_0^{2s})] = \sum_{\sim \in \mathcal{P}(A)} \sum_{t_1, \dots, t_{2s} \in T: t_j \neq t_{j+1}} \sum_{\omega \in \Omega_{\leq}(\sim)} e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})} \prod_{A' \in A/\sim} F_{|A'|}(\tau). \quad (4.8)$$

Ara calcularem

$$I(\sim) = \sum_{\omega \in \Omega_{\leq}(\sim)} e^{\frac{2\pi i}{n} \sum_{j=1}^{2s} \omega_j (t_j - t_{j+1})}.$$

Per cada classe d'equivalència $A' \in A/\sim$ denotarem per $t_{A'}$ l'expressió $t_{A'} := \sum_{a \in A'} (t_a - t_{a+1})$, i per $\omega_{A'}$ l'expressió $\omega_{A'} := \omega_a$ per a cada $a \in A'$ (aquests són tots iguals, ja que $\omega \in \Omega_{\leq}(\sim)$). Llavors

$$I(\sim) = \sum_{(\omega_{A'})_{A' \in A/\sim} \in \mathbb{Z}_n^{|A/\sim|}} e^{\frac{2\pi i}{n} \sum_{A' \in A/\sim} \omega_{A'} t_{A'}} = \prod_{A' \in A/\sim} \sum_{\omega_{A'} \in \mathbb{Z}_n} e^{\frac{2\pi i}{n} \omega_{A'} t_{A'}}.$$

A partir d'aquí veim la importància de (4.8), ja que les sumes de l'interior sumen $|\mathbb{Z}_n| = n$ quan $t_{A'} = 0$ i desapareixen en cas contrari. Llavors, acabem de demostrar el següent resultat.

Lema 4.2.2. *Per a cada classe d'equivalència $A' \in A/\sim$, definim*

$$t_{A'} := \sum_{a \in A'} (t_a - t_{a+1}).$$

Llavors,

$$\mathbb{E}[\text{Tr}(H_0^{2s})] = \sum_{\sim \in \mathcal{P}(A)} \sum_{\substack{t \in T^{2s}: t_j \neq t_{j+1} \\ i \text{ per a tot } A', t_{A'}=0}} n^{|A/\sim|} \prod_{A' \in A/\sim} F_{|A'|}(\tau). \quad (4.9)$$

Aquesta fórmula ens servirà com a base en totes les nostres estimacions. En particular, a partir de la restricció $t_j \neq t_{j+1}$, veim que el sumands s'anul·len si A/\sim conté classes d'equivalència en forma de singletons. Això ens diu de passada que les úniques classes que contribueixen a la suma obeeixen que $|A/\sim| \leq s$.

4.2.3 Estimació clau

Teorema 4.2.3. *Sigui $\tau \leq 1/(1+e)$. Llavors, amb el model de Bernoulli,*

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq \begin{cases} 2 \left(\frac{4}{e(1-\tau)} \right)^s s^{s+1} |\tau n|^s |T|^{s+1}, & s \leq \frac{\tau n}{4(1-\tau)|T|}, \\ \frac{s}{1-\tau} (4s)^{2s-1} |\tau n| |T|^{2s}, & \text{altrament.} \end{cases} \quad (4.10)$$

Demostració. Sigui \sim una classe d'equivalència que no conté cap singletó. Llavors es compleix la següent igualtat.

$$\#\{t \in T^{2s} : t_{A'} = 0 \text{ per a tot } A' \in A/\sim\} \leq |T|^{2s-|A/\sim|+1}.$$

Per veure que és cert basta mirar que, com a combinacions lineals de t_1, \dots, t_{2s} , les expressions $t_j - t_{j-1}$ són totes linealment independents entre sí excepte per la relació $\sum_{j=1}^{2s} t_j - t_{j+1} = 0$. Llavors tenim $|A/\sim| - 1$ restriccions independents en la suma de dalt, i per tant el nombre de t 's que satisfan aquestes restriccions està fitat per $|T|^{2s-|A/\sim|+1}$.

Es segueix llavors de (4.9) i de la fita dels termes individuals de $F_{|A'|}(\tau)$, (2.7), que

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq \sum_{k=1}^s n^k |T|^{2s-k+1} \sum_{\sim \in \mathcal{P}(A)} \prod_{A' \in A/\sim} G(|A'|), \quad (4.11)$$

on $\mathcal{P}(A)$ denota les relacions d'equivalència en A amb k classes d'equivalències i sense singletons. En altres paraules, l'esperança de la traça compleix

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq \sum_{k=1}^s n^k |T|^{2s-k+1} Q(s, k),$$

on

$$Q(s, k) := \sum_{\sim \in \mathcal{P}(A)} \prod_{A' \in A/\sim} G(|A'|).$$

La idea ara és estimar el valor $Q(s, k)$ trobant una desigualtat recursiva. Però abans de fer això, observem que, per $\tau \leq 1/(1+e)$,

$$G(s+1) \leq sG(s),$$

per a tot $s \geq 1$. Per veure això, empram el fet que $\log G$ és convex, i per tant,

$$\log G(s+1) \leq \log G(s) + \frac{d}{ds} \log G(s) \log G(s+1).$$

Llavors, el que volíem veure prové del fet que $\frac{d}{ds} \log G(s) \leq \log s$, sempre que $\log \log \frac{1-\tau}{\tau} \geq 0$.

Ara volem demostrar que la desigualtat

$$Q(s, k) \leq (s-1)Q(s-1, k) + (s-1)G(2)Q(s-2, k-1) \quad (4.12)$$

és vàlida per a tot $s \geq 2$ i $k \geq 1$. Per a veure-ho, suposem que α és un element de A i que \sim és en $\mathcal{P}(s, k)$. Llavors poden passar dues coses:

1. Que α pertanyi a una classe d'equivalència que només conté un altre element β de A (pel qual hi ha $s-1$ opcions), i traient aquesta classe hom obté el terme $(s-1)G(2)Q(s-2, k-1)$.
2. Que α pertanyi a una classe d'equivalència amb més de dos elements, d'on eliminant α de A passa a ser una classe d'equivalència en $\mathcal{P}(A \setminus \{\alpha\}, k)$.

Per a controlar la contribució del cas 2, considerem \sim' un element de $\mathcal{P}(A \setminus \{\alpha\}, k)$ i siguin A_1, \dots, A_k les corresponents classes d'equivalència. L'element α pertany a una de les classes A_i , i provoca que $G(|A_i|)$ creixi com a màxim $|A_i|$. Llavors, la contribució d'aquest terme és més petita que

$$\sum_{\sim' \in \mathcal{P}(A \setminus \{\alpha\}, k)} \sum_{i=1}^k |A_i| \prod_{A' \in A/\sim'} G(|A'|).$$

Però clarament $\sum_{i=1}^k |A_i| = s-1$, i l'expressió es simplifica a $(s-1)Q(s-1, k)$.

De la desigualtat recursiva, hom obté per inducció que

$$G(s, k) \leq G(2)^k (2s)^{s-k}, \quad (4.13)$$

ja que per inducció sobre k és vàlida per a tots els $G(1, k)$'s i $G(2, k)$'s i, si després ho suposem cert per a tots els parells (m, k) amb $m < s$, l'equació (4.12) ens dóna la propietat per al cas $m = s$.

La fita (4.13) es suposa automàticament una fita per a la traça,

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq \sum_{k=1}^s n^k |T|^{2s-k+1} G(2)^k (2s)^{s-k},$$

Si definim β com $\beta := nG(2)/(4s|T|)$, la banda de la dreta es pot reescriure com $|T|^{2s+1}(4s)^{2s} \sum \beta^k$, i pel fet que $\sum \beta^k \leq s \max(\beta, \beta^s)$, acabem de demostrar que

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq s \cdot \begin{cases} n^s |T|^{s+1} G(2)^s (4s)^s, & s \leq \frac{nG(2)}{4|T|}, \\ n |T|^{2s} G(2) (4s)^{2s-1}, & \text{altrament.} \end{cases} \quad (4.14)$$

Recordant que $G(2) = \tau/(1-\tau)$, (4.14) és quasi el contingut del Teorema 4.2.3, excepte per la pèrdua del factor e^s quan n no és massa gran.

Per a recuperar aquest factor addicional, començam observant que (4.12) ens dóna la desigualtat

$$Q(2k, k) \leq (2k-1)G(2)Q(2(k-1), k-1),$$

ja que $Q(s, k) = 0$ per a $s > 2k$. Es segueix que

$$Q(2k, k) \leq (2k-1)(2k-3) \dots 3G(2)^k = \frac{(2k-1)!}{2^{k-1}(k-1)!} G(2)^k,$$

i per simple inducció

$$Q(s, k) \leq (s-1)(s-2) \dots 2k2^{s-k}Q(2k, k) = \frac{(n-1)!}{(k-1)!} 2^{s-2k+1} G(2)^k, \quad (4.15)$$

que és lleugerament millor que (4.13). Resumint,

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq \sum_{k=1}^s B(2s, k), \quad B(2s, k) := \frac{(2s-1)!}{(k-1)!} n^k |T|^{2s-k+1} G(2)^k 2^{2s-2k+1}.$$

Si ara calculam

$$\frac{B(2s, k)}{B(2s, k-1)} = \frac{nG(2)}{4|T|(k-1)},$$

llavors, si fixam s tal que $s \leq nG(2)/(4|T|)$, $B(2s, k)$ és no decreixent respecte k . Per tant,

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq sB(2s, s) = s \frac{2n!}{s!} n^s |T|^{s+1} G(2)^s. \quad (4.16)$$

Notem que emprant la fórmula clàssica de Stirling [Rob67]:

$$\sqrt{2\pi}s^{s+1/2}e^{-s+1/(12s+1)} < s! < \sqrt{2\pi}s^{s+1/2}e^{-s+1/(12s)},$$

hom pot simplificar $(2s)!/s!$ deixant-ho com

$$\frac{2s!}{s!} \leq 2^{2s+1}s^s e^{-s}.$$

Substituint-ho a (4.16) acabam la prova. \square

4.3 Invertibilitat

Teorema 4.3.1. *Sigui T un subconjunt fixat, i triat Ω emprant el model de Bernoulli amb paràmetre τ . Suposem que*

$$|T| \leq C_M \cdot (\log n)^{-1} \cdot \tau n,$$

on C_M és una certa constant. Llavors $\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}$ és invertible amb una probabilitat mínima de $1 - O(n^{-M})$.

Demostració. A partir de la representació del polinomi dual P en (4.1) i de les relacions entre els operadors H i H_0 gràcies al qual hi hem arribat, ens n'adonem que veure que $\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}$ és invertible és equivalent a veure que ho és $I_T - \frac{1}{|\Omega|} H_0$.

Una de les maneres de comprovar-ho és mostrant que l'operador norma, és a dir, el major valor propi, de H_0 és menor que $|\Omega|$. Un pas intermedi en aquest procés és fitar l'operador norma $\|H_0\|$ per la norma de Frobenius, $\|H_0\|_F$:

$$\|H_0\|^2 \leq \|H_0\|_F^2 := \text{Tr}(H_0 H_0^*) = \sum_{t_1, t_2} |(H_0)_{t_1, t_2}|^2, \quad (4.17)$$

on $(H_0)_{t_1, t_2}$ és l'element de la matriu de la fila t_1 i columna t_2 i, a més, es compleix que $|(H_0)_{t_1, t_2}|^2 \sim |\Omega|$.

Llavors, aplicant (4.17), tendriem que seria invertible quan es donàs $|T| \sim |\Omega|$. Per mostrar que $|H_0|$ és relativament petit per a conjunts T grans (tenint en compte que pròximament necessitarem que $|T| \sim |\Omega|(\log n)^{-1}$), emprarem fites de la norma de Frobenius de potències grans de H_0 , beneficiant-nos així dels cancel·laments que es produeixen a raó de l'aleatorietat dels coeficients de la matriu de H_0 .

Considerem que es compleix que $\tau \leq 1/(1+e)$ i $n \leq \tau n/[4|T|(1-\tau)]$. Llavors, emprant la fita clau (4.10), el moment $2s$ de H_0 satisfà

$$\mathbb{E}[\text{Tr}(H_0^{2s})] \leq 2 \left(\frac{4}{e(1-\tau)} \right)^s s^{s+1} |\tau n|^s |T|^{s+1}. \quad (4.18)$$

La fita d'aquest moment ens dóna una estimació de l'operador norma de H_0 . Per veure-ho, notem que, per ser H_0 autoadjunt,

$$\|H_0\|^{2s} = \|H_0^s\|^2 \leq \|H_0^s\|_F^2 = \text{Tr}(H_0^{2s}).$$

Considerant α un nombre positiu entre 0 i 1, es té per la desigualtat de Markov¹ que

$$\mathbb{P}(\|H_0^s\|_F \geq \alpha^s |\tau n|^s) \leq \frac{\mathbb{E}\|H_0^s\|_F^2}{\alpha^{2s} |\tau n|^{2s}}.$$

On si hi apliquem la desigualtat (4.18) tenint en compte que $\|H_0^s\|_F^2 = \text{Tr}(H_0^{2s})$, llavors obtenim

$$\mathbb{P}(\|H_0^s\|_F \geq \alpha^s |\tau n|^s) \leq 2s e^{-n} \left(\frac{4s}{\alpha^2(1-\tau)} \right)^s \left(\frac{|T|}{|\tau n|} \right)^s |T|. \quad (4.19)$$

Cal remarcar que la darrera desigualtat es satisfà per a qualsevol mida $|T|$ del mostreig, amb la condició que $s \leq \tau n/[4|T|(1-\tau)]$.

La prova s'acaba invocant el Teorema 4.3.2, que trobareu a continuació, i que es basa en especialitzar (4.19) per a uns certs $|T|$. \square

Teorema 4.3.2. *Considerem $\tau \leq (1+e)^{-1}$ i suposi's que $|T|$ compleix*

$$|T| \leq \frac{\alpha_M^2(1-\tau)}{4} \frac{|\tau n|}{s}, \text{ per algun } \alpha_M \leq \alpha \leq 1. \quad (4.20)$$

Llavors,

$$\mathbb{P}(\|H_0^s\|_F \geq \alpha^s |\tau n|^s) \leq \frac{1}{2} \alpha^2 e^{-s} |\tau n|.$$

Seleccioni's $s = (M+1) \log n$, el qual es correspon amb les hipòtesis del Teorema 4.3.1. Llavors, l'operador $I_T - \frac{1}{|\Omega|} H_0$ és invertible amb probabilitat, com a mínim, de $1 - 1.25n^{-M}$.

¹*Desigualtat de Markov:* Donada una variable aleatòria X no negativa i un valor $a > 0$, llavors $\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$. En podeu trobar la prova a [Ste05].

Demostració. La primera part del teorema es segueix de (4.19). Per a la segona part, comencem per observar una de les aplicacions típiques de la teoria de grans desviacions [Var84], pel qual

$$\mathbb{P}(|\Omega| < \mathbb{E}[|\Omega|] - t) \leq \exp(-t^2 \mathbb{E}[|\Omega|]),$$

tot i que seria possible trobar millors aproximacions, com en [Bou00]. Del que tenim es segueix que

$$\mathbb{P}(B_M) \leq n^{-M}, \quad B_M := \{|\Omega| < (t - \varepsilon_M)|\tau n|\}, \quad \varepsilon_M := \sqrt{\frac{2M \log n}{|\tau n|}}. \quad (4.21)$$

Ara prenguem $s = (M+1) \log n$ i $\alpha = 1/\sqrt{2}$, i suposem que T satisfà (4.20). Noti's que T satisfà les hipòtesis del Teorema 4.3.1. Si definim $A_M := \{\|H_0\| \geq |\tau n|\sqrt{2}\}$, llavors

$$\mathbb{P}(A_M) \leq \frac{1}{4} |\tau n| n^{-(M+1)} \leq \frac{1}{4} n^{-M},$$

i en el complement de $A_M \cup B_M$,

$$\|H_0\| \leq \tau n / \sqrt{2} \leq |\Omega| / |\sqrt{2}(1 - \varepsilon_M)|.$$

Llavors, segons el raonament fet en l'inici de la prova del Teorema 4.3.1, es té que $I_T - \frac{1}{|\Omega|} H_0$ és invertible i, a més, hem vist que amb la probabilitat desitjada. \square

4.4 Magnitud en T^c

L'objectiu d'aquesta secció és demostrar el Lema 4.4.4. Primer de tot, desenvoluparem una expressió per al polinomi $P(t)$ mitjançant la igualtat algebraica

$$(1 - M)^{-1} = (1 - M^s)^{-1}(1 + M + \dots + M^{s-1}).$$

Per tant, podem escriure,

$$(I_T - \frac{1}{|\Omega|^s} H_0^s)^{-1} = I_T + R, \quad \text{on } R = \sum_{p=1}^{\infty} \frac{1}{|\Omega|^{ps}} H_0^{ps}.$$

Llavors, la inversa ve donada per la sèrie truncada de Neumann

$$(I_T - \frac{1}{|\Omega|} H_0)^{-1} = (I_T + R) \sum_{m=1}^{s-1} \frac{1}{|\Omega|^m} H_0^m. \quad (4.22)$$

El punt clau aquí està en que el reste R és bastant petit en la norma de Frobenius: suposi's que $\|\iota^* H\|_F \leq \alpha|\Omega|$, llavors

$$\|R\|_F \leq \frac{\alpha^s}{1 - \alpha^s}.$$

En particular, els coeficients de la matriu R són individualment més petits que $\alpha^s/(1 - \alpha^s)$. Si introduïm la norma ℓ_∞ d'una matriu com $\|M\|_\infty = \sup_i \|Mx\|_\infty$, la qual també ve donada per

$$\|M\|_\infty = \sup_i \sum_j |M(i, j)|,$$

és segueix de la desigualtat de Cauchy-Schwarz [Ste05] que

$$\|M\|_\infty^2 = \sup_i \# \text{col}(M) \sum_j |M(i, j)|^2 \leq \# \text{col}(M) \cdot \|M\|_F^2,$$

on $\# \text{col}(M)$ és el nombre de columnes de M . Aquesta observació ens dona la fita

$$\|R\|_\infty \leq |T|^{1/2} \frac{\alpha^s}{1 - \alpha^s}. \quad (4.23)$$

Com veure a continuació, la fita (4.23) fa que puguem ometre el terme R de (4.22). L'única dificultat ara és establir una bona fita per a la sèrie de Neumann

$$\frac{1}{|\Omega|} H \sum_{m=1}^{s-1} \frac{1}{|\Omega|^m} H_0^m.$$

Fites de sèries truncades de Neumann

A partir de (4.1) observam que, en el complement de T ,

$$P = \frac{1}{|\Omega|} H (I_T - \frac{1}{|\Omega|} H_0)^{-1} \iota^* \text{sgn}(f),$$

ja que la component ι de (4.1) s'anul·la al complement de T . Aplicant (4.22), podem reescriure P com

$$P(t) = P_0(t) + P_1(t), \quad \forall t \in T^c,$$

on

$$P_0(t) = S_s \text{sgn}(f)(t), \quad P_1(t) = \frac{1}{|\Omega|} H R \iota^* (I + S_{s-1}) \text{sgn}(f)(t)$$

i

$$S_s = \sum_{m=1}^s |\Omega|^{-m} (H\iota^*)^m.$$

Siguin $a_0, a_1 > 0$ dos nombres tals que sumen 1. Llavors,

$$\mathbb{P}\left(\sup_{t \in T^c} |P(t)| < 1\right) \leq \mathbb{P}(\|P_0\|_\infty > a_0) + \mathbb{P}(\|P_1\|_\infty > a_1),$$

i la idea és fitar cada terme individualment. Definim $Q_0 := S_{s-1} \operatorname{sgn}(f)$. Així, $P_1 = \frac{1}{|\Omega|} HR\iota^*(\operatorname{sgn}(f) + Q_0)$. Amb aquestes notacions observam que

$$\|P_1\|_\infty \leq \frac{1}{|\Omega|} \|HR\|_\infty (1 + \|\iota^* Q_0\|_\infty).$$

Per tant, la fita per a la magnitud de P_1 provindrà de la fita en $\|HR\|_\infty$ juntament amb la fita per a la magnitud de $\iota^* Q_0$. Serà suficient trobar fites per a $\|Q_0\|_\infty$ (ja que $\|\iota^* Q_0\|_\infty \leq \|Q_0\|_\infty$), que provindran de fites sobre P_0 , ja que són quasi iguals amb Q_0 .

Fixem $t \in T^c$ i escriguem $P_0(t)$ com

$$P_0(t) = \sum_{m=1}^s |\Omega|^{-m} X_m(t), \quad X_m = (H\iota^*)^m \operatorname{sgn}(f).$$

La idea és emprar l'estimació de moments per a controlar la grandària de cada terme $X_m(t)$.

Lema 4.4.1. *Sigui $s = km$. Llavors, $\mathbb{E}|X_m(t_0)|^{2k}$ satisfà la mateixa estimació que en el Teorema 4.2.3 (amb un factor multiplicatiu $|T^{-1}|$), és a dir,*

$$\mathbb{E}|X_m(t_0)|^{2k} \leq \frac{1}{|T|} B_s,$$

on B_s és la part dreta de l'equació (4.10). En particular, a partir de (4.18) es té que

$$\mathbb{E}[\operatorname{Tr}(H_0^{2s})] \leq 2 \left(\frac{4}{e(1-\tau)} \right)^s s^{s+1} |\tau n|^s |T|^s, \quad (4.24)$$

en tant que $s \leq \frac{\tau n}{4(1-\tau)|T|}$.

Demostració. Per abreviar, diguem $e^{i\phi} = \operatorname{sgn}(f)$. Fixem K . Emprant (4.4), tenim

$$[(H\iota^*)^{s+1} e^{i\phi}](t_0) = \sum_{t_1, \dots, t_s \in T: t_j \neq t_{j+1} \text{ for } j=0, \dots, s} \sum_{\omega_0, \dots, \omega_s \in \Omega} e^{\frac{2\pi i}{n} \sum_{j=0}^s \omega_j (t_j - t_{j+1})} e^{i\phi(t_{s+1})},$$

i, per exemple,

$$\begin{aligned} |[(Ht^*)^{s+1}e^{i\phi}](t_0)|^2 &= \sum_{\substack{t_1, \dots, t_s \in T: t_j \neq t_{j+1} \text{ for } j=0, \dots, s \\ t'_1, \dots, t'_s \in T: t'_j \neq t'_{j+1} \text{ for } j=0, \dots, s}} e^{i\phi(t_n+1)} e^{-i\phi(t'_n+1)} \\ &\times \sum_{\substack{\omega_0, \dots, \omega_s \in \Omega \\ \omega'_0, \dots, \omega'_s \in \Omega}} e^{\frac{2\pi i}{n} \sum_{j=0}^s \omega_j (t_j - t_{j+1})} e^{-\frac{2\pi i}{n} \sum_{j=0}^s \omega'_j (t'_j - t'_{j+1})}. \end{aligned}$$

Llavors, hom calcula el moment $2K$ -èssim d'una manera semblant. Sigui $m := K(s+1)$ i

$$\boldsymbol{\omega} := (\omega_j^{(k)})_{k,j}, \quad \mathbf{t} = (t_j^{(k)})_{k,j} \in T^{2K(s+1)}, \quad 1 \leq j \leq s+1 \text{ i } 1 \leq k \leq 2K.$$

Amb aquesta notació, podem escriure

$$|[(Ht^*)^{s+1}e^{i\phi}](t_0)|^{2K} = \sum_{\mathbf{t} \in T^{2m}: t_j^{(k)} \neq t_{j+1}^{(k)}} \sum_{\boldsymbol{\omega} \in \Omega^{2m}} e^{iS_1} e^{\frac{2\pi i}{n} S_2},$$

on $S_1 = \sum_{k=1}^{2K} (-1)^k \phi(t_{s+1}^{(k)})$, $S_2 = \sum_{k=1}^{2K} \sum_{j=0}^s (-1)^k \omega_j^{(k)} (t_j^{(k)} - t_{j+1}^{(k)})$, i on hem emprat la convenció que $x_0^{(k)} = x_0$ per a tot $1 \leq k \leq 2K$, i on s'ha d'entendre que la condició $t_j^{(k)} - t_{j+1}^{(k)}$ és vàlida per $0 \leq j \leq s$.

Ara, els càlculs de l'esperança van exactament com en la secció 4.2. És a dir, definim una relació d'equivalència \sim en un conjunt finit $A := \{0, \dots, s\} \times \{1, \dots, 2K\}$ imposant que $(j, k) \sim (j', k')$ si $\omega_j^{(k)} = \omega_{j'}^{(k')}$ i observant com abans que

$$\mathbb{E} \left[\prod_{j,k} I_{\omega_j^{(k)}} \right] = \tau^{|A/\sim|};$$

això és, τ elevada a la potència igual al nombre de ω 's distints i, llavors, podem escriure el valor esperat $m(s; K)$ com

$$\begin{aligned} m(s; K) &= \sum_{\mathbf{t} \in T^{2m}: t_j^{(k)} \neq t_{j+1}^{(k)}} e^{i \sum_{k=1}^{2K} (-1)^k \phi(t_{s+1}^{(k)})} \sum_{\sim \in \mathcal{P}(A)} \tau^{|A/\sim|} \\ &\times \sum_{\boldsymbol{\omega} \in \Omega(\sim)} e^{\frac{2\pi i}{n} \sum_{k=1}^{2K} \sum_{j=0}^s (-1)^k \omega_j^{(k)} (t_j^{(k)} - t_{j+1}^{(k)})}. \end{aligned}$$

Com abans, mitjançant el Lema 4.2.2 i reagrupant-ho obtenim

$$\begin{aligned} m(s; K) &= \sum_{\sim \in \mathcal{P}(A)} \sum_{\mathbf{t} \in T^{2m}: t_j^{(k)} \neq t_{j+1}^{(k)}} e^{i \sum_{k=1}^{2K} (-1)^k \phi(t_{s+1}^{(k)})} \prod_{A' \in A/\sim} F_{|A'|}(\tau) \\ &\times \sum_{\boldsymbol{\omega} \in \Omega(\sim)} e^{\frac{2\pi i}{n} \sum_{k=1}^{2K} \sum_{j=0}^s (-1)^k \omega_j^{(k)} (t_j^{(k)} - t_{j+1}^{(k)})}. \end{aligned}$$

Igual que férem amb anterioritat, el sumatori variant ω s'anul·la excepte que $t'_A := \sum_{(j,k) \in A'} (-1)^k (t_j^{(k)} - t_{j+1}^{(k)}) = 0$ per totes les classes d'equivalència $A' \in A/\sim$, i en tal cas la suma resultant és igual a $N^{|A/\sim|}$. En particular, si A/\sim , llavors la suma desapareix a raó de les restriccions $t_j^{(k)} \neq t_{j+1}^{(k)}$, així que només haurem de restringir a les classes que no contenguin singletons. En particular, tenim que

$$|A/\sim| \leq K(s+1) = m.$$

Resumint,

$$\begin{aligned} m(s; K) &= \sum_{\sim \in \mathcal{P}(A)} \sum_{\mathbf{t} \in T^{2m} : t_j^{(k)} \neq t_{j+1}^{(k)} \text{ i } t_{A'} = 0 \ \forall A'} e^{i \sum_{k=1}^{2K} (-1)^k \phi(t_{s+1}^{(k)})} N^{|A/\sim|} \prod_{A' \in A/\sim} F_{|A'|}(\tau) \\ &\leq \sum_{\sim \in \mathcal{P}(A)} \sum_{\mathbf{t} \in T^{2(s+1)} : t_j^{(k)} \neq t_{j+1}^{(k)} \text{ i } t_{A'} = 0 \ \forall A'} N^{|A/\sim|} \prod_{A' \in A/\sim} F_{|A'|}(\tau), \end{aligned}$$

ja que $|e^{i \sum_{k=1}^{2K} (-1)^k \phi(t_{s+1}^{(k)})}| = 1$. Noti's la semblança amb (4.9). Sigui \sim una equivalència que no conté cap singleton. Llavors, es compleix

$$\#\{\mathbf{t} \in T^{2K(s+1)} : t_{A'} = 0, \text{ per a tot } A' \in A/\sim\} \leq |T|^{2K(s+1) - |A/\sim|}.$$

Per veure que això és cert, vegem-ho com a combinacions lineals dels $t_j^{(k)}$ i de t_0 . Notem que les expressions $t_j^{(k)} - t_{j+1}^{(k)}$ són totes linealment independents, i per tant les expressions $\sum_{(j,k) \in A} t_j^{(k)} - t_{j+1}^{(k)}$ també. Llavors, tenim $|A/\sim|$ restriccions independents en la suma anterior, i a més el nombre de t 's que satisfan les restriccions està fitat per $|T|^{2K(s+1) - |A/\sim|}$. Amb les notacions de la secció 4.2, hem demostrat

$$m(s, K) \leq \sum_{k=1}^m n^k |T|^{2m-k} \sum_{\sim \in \mathcal{P}(A, k)} \prod_{A' \in A/\sim} G(|A'|),$$

que és el mateix que (4.11), pel qual hem provar que es satisfà la fita desitjada. \square

Lema 4.4.2. *Fixem $a_0 = 0.91$. Suposem que $|T|$ satisfà (4.20) i sigui B_M el conjunt on $|\Omega| < (1 - \varepsilon_M)(\tau n)$, amb ε_M com en (4.21). Per a cada $t \in \mathbb{Z}_n$, hi ha un conjunt A_t amb la propietat*

$$\mathbb{P}(A_t) > 1 - \varepsilon_s, \quad \varepsilon_s = 2(1 - \varepsilon_M)^{-2s} s^2 e^{-s} \alpha^2 (0.42)^{-2s},$$

i

$$|P_0(t)| < 0.91, \quad |Q_0(t)| < 0.91 \quad \text{en } A_t \cap B_M^c.$$

Com a conseqüència,

$$\mathbb{P}(\sup_t |P_0(t)| > a_0) \leq n^{-M} + n\varepsilon_s,$$

i de manera similar per Q_0 .

Demostració. Suposem que s és de la forma $s = 2^J - 1$, cosa que ens simplificarà la prova. Per a cada m i cada k tal que $km \geq s$, es té per (4.20), (4.24) i uns quants càlculs senzills que

$$\mathbb{E}|X_m(t)|^{2k} \leq 2se^{-s}\alpha^{2s}|\tau n|^{2s}. \quad (4.25)$$

De nou tenim que $|\Omega| \approx |\tau n|$, i buscarem una fita en el conjunt B_M^c , on $|\Omega| \geq (1 - \varepsilon_M)|\tau n|$. En aquest conjunt,

$$|P_0(t)| \leq \sum_{m=1}^s Y_m, \quad Y_m = \frac{1}{(1 - \varepsilon_M)^m |\tau n|^m} |X_m(t)|.$$

Fixem $\beta_j > 0$, per $0 \leq j < J$, tal que $\sum_{j=0}^{J-1} 2^j \beta_j \leq a_0$. Òbviament,

$$\mathbb{P}\left(\sum_{m=1}^s Y_m > a_0\right) \leq \sum_{j=0}^{J-1} \sum_{m=2^j}^{2^{j+1}-1} \mathbb{P}(Y_m > \beta_j) < \sum_{j=0}^{J-1} \sum_{m=2^j}^{2^{j+1}-1} \beta_j^{2K_j} \mathbb{E}|Y_m|^{2K_j},$$

on $K_j = 2^{J-j}$. Observi's que, per a cada m tal que $2^j \leq m < 2^{j+1}$, $K_j m$ satisfà $s \leq K_j m < 2s$ i, per tant, (4.25) dóna

$$\mathbb{E}|Y_m|^{2K_j} \leq (1 - \varepsilon_M)^{-2s} (2se^{-s}\alpha^{2s}).$$

Per exemple, prenent $\beta_j^{-K_j}$ constant per a tot j , és a dir, igual a β_0^{-s} , tendríem

$$\mathbb{P}\left(\sum_{m=1}^s Y_m > a_0\right) \leq 2(1 - \varepsilon_M)^{-2s} s^2 e^{-s} \alpha^{2s} \beta_0^{-2s},$$

amb $\sum_{j=0}^{J-1} 2^j \beta_j \leq a_0$. A partir de càlculs purament numèrics es pot comprovar que, per a $\beta_0 = 0.42$, $\sum_{j=0}^{J-1} 2^j \beta_j \leq 0.91$, que dóna

$$\mathbb{P}\left(\sum_{m=1}^s Y_m > 0.91\right) \leq 2(1 - \varepsilon_M)^{-2s} s^2 e^{-s} \alpha^{2s} (0.42)^{-2s}.$$

Per Q_0 es segueix exactament el mateix procediment, acabant la prova. \square

Lema 4.4.3. *Fixem $a_1 = 0.09$. Suposem que el parell (α, s) satisfà $|T|^{3/2} \frac{\alpha^s}{1-\alpha^s} \leq a_1/2$. Llavors,*

$$\|P_1\|_\infty \leq a_1$$

en l'esdeveniment $A \cap \{\|l^ H\|_F\} \leq \alpha|\Omega|\}$, per qualche A tal que $\mathbb{P}(A) \geq 1 - O(n^{-M})$.*

Demostració. Així com hem vist abans,

1. $\|P_1\|_\infty \leq \|H\|_\infty \|R\|_\infty (1 + \|Q_0\|_\infty)$,
2. Q_0 satisfà la fita del Lema 4.4.2.

Considerem l'esdeveniment $\{\|Q_0\|_\infty \leq 1\}$. En ell, $\|P_1\| \leq a_1$ si $\frac{1}{|\Omega|} \|H\| \|R\|_\infty \leq a_1/2$. La matriu H satisfà $\frac{1}{|\Omega|} \|H\|_\infty \leq |T|$, ja que H té $|T|$ columnes i cada element de la matriu està fitat, de manera un poc rude, per $|\Omega|$. Es segueix doncs de (4.23) que

$$\|H\|_\infty \|R\|_\infty \leq |T|^{3/2} \frac{\alpha^s}{1 - \alpha^s},$$

amb probabilitat de com a mínim $1 - O(n^{-M})$. Llavors, només resta triar un α i un s tal que la banda de la dreta sigui menor de $a_1/2$. \square

El lema

Lema 4.4.4. *Sota les hipòtesis del Teorema 4.3.1, P definit a (4.1) satisfà $|P(t)| < 1$, per a tot $t \in T^c$, amb una probabilitat mínima de $1 - O(n^{-M})$.*

Demostració. Fixam $M > 0$. Elegim $\alpha = 0.42(1 - \varepsilon_M)$, on ε_M està pres com en (4.21), i s com l'enter més proper a $(M + 1) \log n$. Amb aquesta elecció tenim el següent:

1. $\varepsilon_s = 2[(M + 1) \log n]^2 n^{-(M+1)}$, i llavors, el Lema 4.4.2 implica que tant P_0 com Q_0 estan fitats per 0.91 a fora de T^c amb una probabilitat mínima de $1 - [1 + 2((M + 1) \log n)^2] n^{-M}$.
2. El Lema 4.4.3 assegura que és suficient tenir $n^{3/2} \alpha^s / (1 - \alpha^s) \leq 0.045$ per a que es compleixi $|P_1(t)| < 0.09$ en T^c . Com que $\log(0.42) \approx -0.87$, i $\log(0.045) \approx -3.10$, aquesta condició és aproximadament equivalent a

$$(1.5 - 0.87(M + 1)) \log n \leq -3.10.$$

Prenent, per exemple, $M \leq 2$, llavors l'anterior desigualtat es compleix per $n \geq 17$.

Per acabar, el resultat del Lema 4.4.4 es dona amb probabilitat, com a mínim, de $1 - O([(M + 1) \log n]^2) n^{-M}$ si T compleix

$$|T| \leq C_M \frac{|\tau n|}{\log n}, \quad C_M = \frac{0.42^2(1 - \tau)}{4(M + 1)} (1 + O(1)). \quad (4.26)$$

\square

Capítol 5

Recuperació probable de senyals

5.1 Recuperació probablement exacta de senyals

Considerem una senyal $f(t) \in \mathbb{R}^n$ de la qual volem obtenir els n coeficients però només en tenim m de la forma (3.1) on $m < n$.

Amb aquesta informació, decidim recuperar la senyal f minimitzant en norma ℓ_1 . La reconstrucció proposada serà f^* , que ve donat per $f^* = \Psi x^*$, on x^* és la solució del programa d'optimització convexa

$$\min_{\tilde{x} \in \mathbb{R}^n} \|\tilde{x}\|_{\ell_1}, \quad \text{subjecte a} \quad y_k = \langle \varphi_k, \Psi \tilde{x} \rangle, \quad k = 1, \dots, m. \quad (5.1)$$

Això és, de tots els valors de $\tilde{f} = \Psi \tilde{x}$ consistents amb les dades de la mostra, prenem la solució que minimitzi la norma ℓ_1 .

L'ús de la norma ℓ_1 com a funció promotora de la dispersió prové de dècades enrere, amb un primera aplicació en sismologia. Tot i així, aquesta no és la única manera de recuperar senyals disperses; un altre mètode proposat és l'algorisme voraç¹.

Sabem que almenys existeix una solució de (5.1). Però el que no està massa clar encara és el fet que aquesta solució sigui única i, a més, que sigui la que estam cercant. Suposem que f té suport en T i que nosaltres observam \hat{f} en Ω . El següent

¹Un algorisme voraç és aquell que, per a resoldre un determinat problema, segueix una heurística consistent en triar l'opció òptima en cada pas local amb l'esperança d'arribar a una solució general òptima.

resultat ens mostra que una condició necessària i suficient per a que f sigui solució de (5.1) és l'existència d'un polinomi trigonomètric P amb unes determinades propietats.

Lema 5.1.1. *Sigui $\Omega \subset \mathbb{Z}_n$. Per a un vector $f \in \mathbb{C}^n$ amb $T := \text{supp}(f)$, definim la funció $\text{sgn}(f)$ com en la secció 4.1. Suposem que existeix un vector P la transformada de Fourier del qual, \hat{P} , té suport en Ω tal que satisfà*

$$P(t) = \text{sgn}(f)(t) \text{ per a tot } t \in T \quad (5.2)$$

i

$$|P(t)| < 1 \text{ per a tot } t \in T^c. \quad (5.3)$$

Llavors, si $\mathcal{F}_{t \rightarrow \Omega}$ és injectiva, el minimitzador $f^\#$ del problema (5.1) és únic i igual a f . Per altra banda, si f és l'únic minimitzador del problema (5.1), llavors existeix un vector P que compleix les propietats anteriors.

Demostració. Suposem Ω no buid i f no idènticament zero, ja que si no el resultat és trivial.

Suposem primer que la funció P existeix. Sigui g qualsevol vector distint a f però tal que $\hat{g}|_\Omega = \hat{f}|_\Omega$. Escriguem $h := f - g$, d'on \hat{h} s'anul·larà a Ω . Per a qualsevol $t \in T$ tenim

$$\begin{aligned} |g(t)| &= |f(t) + h(t)| \\ &= ||f(t)| + h(t)\overline{\text{sgn}(f)(t)}| \\ &\geq |f(t)| + \Re(h(t)\overline{\text{sgn}(f)(t)}) \\ &= |f(t)| + \Re(h(t)\overline{P(t)}), \end{aligned}$$

mentre que per a $t \notin T$ tenim $|g(t)| \geq |f(t)| \geq \Re(h(t)\overline{P(t)})$, ja que $|P(t)| < 1$. Llavors,

$$\|g\|_{\ell_1} \geq \|f\|_{\ell_1} + \sum_{t=0}^{n-1} \Re(h(t)\overline{P(t)}).$$

Però la fórmula de Parseval ens dóna

$$\sum_{t=0}^{n-1} \Re(h(t)\overline{P(t)}) = \frac{1}{n} \sum_{k=0}^{n-1} \Re(\hat{h}(k)\overline{\hat{P}(k)}) = 0,$$

ja que \hat{P} està suportada dins Ω i \hat{h} s'anul·la en Ω . Llavors, $\|g\|_{\ell_1} \geq \|f\|_{\ell_1}$. Vegem ara quan és que es dóna la igualtat. Si ens hi fixem, l'argument anterior força a que $|h(t)| = \Re(h(t)\overline{P(t)})$ per a tot $t \notin T$. Com que $|P(t)| < 1$, això força a que

$h(t)$ s'anul·li fora de T . Com que \hat{h} s'anul·la en Ω , llavors, per la injectivitat de $\mathcal{F}_{T \rightarrow \Omega}$, veim que h s'ha d'anul·lar quasi per tot i, per tant, $g = f$. Això prova que f és l'únic minimitzador, $f^\#$, del problema (5.1).

Per altra banda, suposem que $f = f^\#$ és l'únic minimitzador del problema (5.1). Sense pèrdua de generalitat, el podem suposar normalitzat, és a dir, $\|f\|_{\ell_1} = 1$. Llavors, la bolla unitat tancada, $B := \{g : \|g\|_{\ell_1} \leq 1\}$ i l'espai afí $V := \{g : \hat{g} \cdot \Omega = \hat{f} \cdot \Omega\}$ intersecten exactament en un punt, que és f .

Pel Teorema de Hahn-Banach [Rud91], podem trobar una funció P tal que l'hiperplà

$$\Gamma_1 := \{g : \sum \Re(g(t)\overline{P(t)}) = 1\}$$

conté V , i tal que el l'espai meitat,

$$\Gamma_{\leq 1} := \{g : \sum \Re(g(t)\overline{P(t)}) \leq 1\},$$

conté B . Transformant adequadament l'espai si és necessari, i emprant la unicitat de la intersecció entre B i V , podem suposar que $\Gamma_1 \cap B$ està contingut dins el subespai minimal que conté f , és a dir, $\{g \in B : \text{supp}(g) \subseteq T\}$.

Com que B està dins $\Gamma_{\leq 1}$, veim que $\sup_t |P(t)| \leq 1$; com que $f \in \Gamma_1 \cap B$, tenim que $P(t) = \text{sgn}(f)(t)$ quan $t \in \text{supp}(f)$. Com que $\Gamma_1 \cap B$ està contingut dins subespai minimal de B que conté f , observem que $|P(t)| < 1$ quan $t \notin T$. Com que Γ_1 conté V , tenim per la identitat de Parseval que \hat{P} té suport dins Ω . Per tant, el resultat es compleix. \square

Una de les preguntes clau era quin era el nombre mínim de mostres per tal de recuperar la senyal. En el cas de senyals disperses com de les que hem parlat en el capítol 3 tenim el Teorema 5.1.5, però abans necessitem el Lema 5.1.4. Per demostrar-lo necessitarem els dos teoremes següents.

Teorema 5.1.2. *Siguin $\{m_n\}$, $\{m_n p_n\}$ i $\{m_n(1 - p_n)\}$, per $n = 1, 2, \dots$, sèries no decreixents d'enters positius amb els m_n estrictament creixents i els p_n entre 0 i 1. Llavors, la sèrie $\{B(m_n p_n + r; m_n, p_n)\}$ és estrictament creixent (decreixent) si r és un enter negatiu (no negatiu).*

Demostració. La sèrie $B(m_n p_n + r; m_n, p_n)$ pot ser interpretada com a la probabilitat d'obtenir menys de $(m_{n+1} p_{n+1} + r)$ encerts en m_{n+1} assajos en els quals m_n probabilitats de succés són iguals a p_n , $m_{n+1} p_{n+1} - m_n p_n$ dels quals són iguals a 1 i els restants $m_{n+1}(1 - p_{n+1}) - m_n(1 - p_n)$ són iguals a 0.

Per completar la prova només necessitam aplicar el Teorema de Hoeffding [Hoe63], el que, en part, diu que si el valor mitjà dels successos en n assajos independents és λ , llavors la probabilitat d'encertar menys o igual a k vegades és \geq (respectivament, \leq) $B(k; n, \lambda/n)$ si $k \geq \lambda$ (respectivament, $k \leq \lambda - 1$), i que la igualtat passa només si els n successos de probabilitats són iguals a λ/n . \square

Teorema 5.1.3. *Sigui t un enter fixat positiu i major a 1. Llavors, així com $n \rightarrow \infty$, $B(n - 1, nt, 1/t)$ creix cap a $\frac{1}{2}$ així com $B(n; nt, 1/t)$ decreix cap a $\frac{1}{n}$.*

Demostració. Pel Teorema del Límit Central [Cra37], ambdós límits són $\frac{1}{2}$. La monoticitat prové d'aplicar el Teorema 5.1.2, amb $p_n = 1/t$, $m_n = nt$ i $r = -1$ per a la primera sèrie i amb $r = 0$ per a la segona. \square

El lema afirma el següent:

Lema 5.1.4. *Si el nombre mitjà d'encerts en n assajos és un enter k , llavors la mediana és també k .*

Demostració. Primer suposem que totes les probabilitats d'encert són igual a k/n . Pel Teorema 5.1.2, la sèrie $\{B(k - 1; n, k/n)\}$ i $\{B(k; n, k/n)\}$ són, respectivament creixents i decreixents en n , tal que

$$B(k - 1; n, k/n) < B(k - 1; kn, 1/n) < \frac{1}{2} < B(k; kn, 1/n) < B(k; n, k/n),$$

d'on les dues desigualtats del centre es segueixen del Teorema 5.1.3.

Si les probabilitats de succés són distintes, aplicarien el Teorema de Hoeffding com en la prova del Teorema 5.1.2 per concloure que

$$\mathbb{P}(\leq k - 1 \text{ encerts}) < B(k - 1; n, k/n) < \frac{1}{2} < B(k; n, k/n) < \mathbb{P}(\leq k \text{ encerts}).$$

\square

Teorema 5.1.5. *Sigui $f \in \mathbb{R}^n$ fixat i suposem que els coeficients x de f en la base Ψ són S -dispersos. Seleccionats m mesuraments en el domini Φ de forma uniformement aleatòria, llavors si*

$$m \geq C_M \cdot \mu^2(\Psi, \Phi) \cdot S \cdot \log n,$$

per alguna constant positiva C_M dependent d'una constant de precisió M , la solució de (5.1) és única i igual a f amb una probabilitat de $1 - O(n^{-M})$.

Demostració. El Lema 5.1.1 ens dóna l'existència i unicitat de solució del problema (5.1) condicionat a l'existència d'un polinomi amb unes certes propietats. Amb les hipòtesis del Teorema 5.1.5, i prenent C_M com en (4.26), el Lema 4.3.1 i el Lema 4.4.4 ens asseguren que aquest polinomi, definit com en (4.1) existeix amb una probabilitat com a mínim de $1 - O(n^{-M})$. Per tant, el resultat queda demostrat si tenim en compte que prenem les mostres segons el model de Bernoulli.

Falta estendre-ho per al model uniforme, que és el que ens interessa.

Definim $\text{Fracàs}(\Omega_0)$ com l'esdeveniment on cap polinomi dual P , suportat en Ω_0 en l'espai de Fourier, existeix tal que compleix les propietats (5.2) i (5.3). Sigui Ω de cardinal m presa seguint el model uniforme, i sigui Ω' presa seguint el model de Bernoulli amb paràmetre $\tau = m/n$. Tenim que

$$\begin{aligned} \mathbb{P}(\text{Fracàs}(\Omega')) &= \sum_{k=0}^n \mathbb{P}(\text{Fracàs}(\Omega') \mid |\Omega'| = k) \cdot \mathbb{P}(|\Omega'| = k) \\ &= \sum_{k=0}^n \mathbb{P}(\text{Fracàs}(\Omega_k)) \cdot \mathbb{P}(|\Omega'| = k), \end{aligned}$$

on Ω_k està seleccionat de manera uniformement aleatòria amb $|\Omega_k| = k$. Notem que:

- $\mathbb{P}(\text{Fracàs}(\Omega_k))$ és una funció no creixent en k . Aquest fet prové directament de

$$\Omega_1 \subset \Omega_2 \quad \Rightarrow \quad \mathbb{P}(\text{Fracàs}(\Omega_1)) \leq \mathbb{P}(\text{Fracàs}(\Omega_2)).$$

Per tant, com major és Ω , més fàcil ens resulta construir un P vàlid.

- Com que τn és un enter, és la mitjana de $|\Omega'|$, es dedueix del Lema 5.1.4 que

$$\mathbb{P}(|\Omega'| \leq \tau n - 1) < 1/2 < \mathbb{P}(|\Omega'| \leq \tau n).$$

Per tant, si tenim això en compte,

$$\begin{aligned} \mathbb{P}(\text{Fracàs}(\Omega')) &\geq \sum_{k=0}^m \mathbb{P}(\text{Fracàs}(\Omega_k)) \cdot \mathbb{P}(|\Omega'| = k) \\ &\geq \mathbb{P}(\text{Fracàs}(\Omega)) \cdot \sum_{k=0}^m \mathbb{P}(|\Omega'| = k) \\ &\geq \frac{1}{2} \cdot \mathbb{P}(\text{Fracàs}(\Omega)). \end{aligned}$$

Concloent, si podem fitar la probabilitat de fracàs per al model de Bernoulli, també podem fitar la taxa de fracàs per al model uniforme, el qual serà de com a molt el doble que la de Bernoulli. \square

Segons els càlculs de Candès, Romberg i Tao [Can04b], per a un valor d'incoherència màxima, és a dir, $\mu^2(\Psi, \Phi) = 1$, el valor per a la quantitat C_M és aproximadament de $23(M + 1)$. Aquest fet està garantit per a $|\Omega| \leq n/4$, $m \geq 2$ i $n \geq 20$. Per tant, com que $\mu(\Psi, \Phi) \in [1, \sqrt{n}]$, llavors $\mu^2(\Psi, \Phi) \in [1, n]$. D'aquesta manera, noti's que la condició del Teorema 5.1.5 compleix

$$m \geq C_M \cdot \mu^2(\Psi, \Phi) \cdot S \cdot \log n \geq C_M \cdot S \cdot \log n.$$

Observació 5.1.6. Cal observar tres fets molt importants:

1. El rol que juga la incoherència és totalment transparent: com menor és la coherència menor nombre de mostres necessitam.
2. No tenim pèrdua d'informació pel fet d'agafar un nombre de mostres m molt menor de n . Si $\mu(\Psi, \Phi)$ és proper a 1, llavors és suficient un nombre de mostres de l'ordre de $S \log n$.
3. La senyal f pot ser recuperada a partir d'un conjunt de dades molt petit minimitzant un funcional convex el qual no pressuposa cap coneixement sobre el nombre de coordenades de x distintes a zero, la seva localització, ni la seva amplitud.

Segons el Teorema 5.1.5, existeixen algunes funcions per a les quals no existeixen conjunts de m mostres tal que la funció f pugui ser recuperada minimitzant la norma ℓ_1 , fins i tot si el suport de f té un cardinal molt menor a n . A continuació en presentam un exemple.

Exemple 5.1.7. Suposem n un quadrat perfecte i considerem una senyal basada en pics d'altura 1 i equidistants amb una separació de \sqrt{n} , la podem veure a la imatge 5.1. Aquesta senyal, que podem escriure com a $\Delta_T = \sum_{k=0}^{\infty} \delta(t - kT)$, és utilitzada com a extrem en les provatures de principis d'incertesa, ja que una de les seves propietats remarcables és que resulta ser invariant respecte a la transformada de Fourier.

Llavors, suposem Ω el conjunt de totes les freqüències tret de les múltiples de n , concretament es té, $m = |\Omega| = n - \sqrt{n}$. Llavors, $\hat{f}|_{\Omega} = 0$ i, òbviament, la reconstrucció obtinguda és zero.

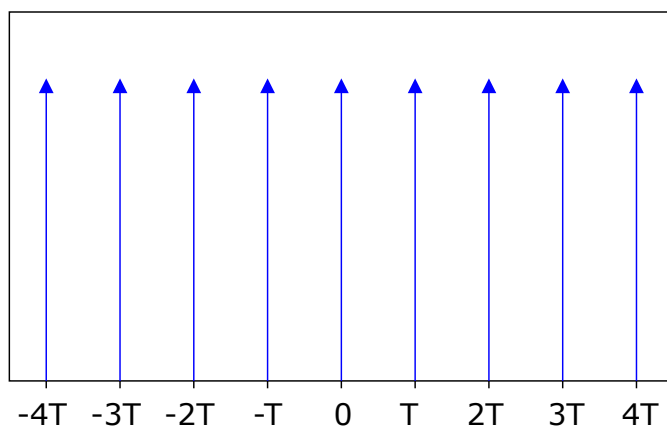


Figura 5.1: *Dirac discrete comb*. En el nostre cas, $T = \sqrt{n}$ i altura 1.

De fet, noti's com el problema d'aquest exemple no cau directament en la minimització ℓ_1 , sinó en el fet que f no pot ser reconstruïda a partir de les seves mostres de Fourier en Ω mostrant que el Teorema 5.1.5 no funciona per a nombres aleatoris de mostres.

5.2 Límit fonamental

Del Teorema 5.1.5 se n'extreu que per a cada senyal f amb suport en un conjunt arbitrari T tal que $|T| = S$ en el domini de temps, llavors (5.1) recupera f exactament –amb una probabilitat alta– a partir d'un nombre de mostres de freqüències superior a $K_M S \log(n)$, on $K_M = C_M \cdot \mu^2(\Psi, \Phi)$. És natural voler estudiar quin és el límit fonamental. En altres paraules, pot un algorisme recuperar una senyal arbitrària a partir d'un nombre molt petit d'observacions aleatòries i amb la mateixa probabilitat de reconstrucció exacta?

Està clar que el nombre de mostres almenys ha de ser proporcional a S , ja que si no $\mathcal{F}_{T \rightarrow \Omega}$ no seria injectiva. De fet, ara veurem com també ha de ser proporcional $K_M \log(n)$ per a garantir la recuperació de determinades senyals a partir d'una gran majoria de conjunts Ω d'una certa mesura.

Suposem $f = \Delta_{\sqrt{n}}$, que és la senyal de l'exemple 5.1.7. Si volem tenir oportunitat de recuperar f , llavors, com a mínim, el conjunt d'observacions Ω i el suport de l'espai de freqüències $W = \text{supp } \hat{f}$ han de coincidir en almenys un punt, altrament, ens remetem al cas de l'exemple esmentat. Triant Ω aleatòriament amb una

distribució uniforme, la probabilitat de que no inclogui cap membre de W és

$$\mathbb{P}(\Omega \cap W = \emptyset) = \frac{\binom{n-\sqrt{n}}{m}}{\binom{n}{m}} \geq \left(1 - \frac{2m}{n}\right)^{\sqrt{n}},$$

on suposem que $m > S = \sqrt{n}$. Llavors perquè $\mathbb{P}(\Omega \cap W = \emptyset)$ sigui més petit que n^{-K_M} , s'ha de donar que

$$\sqrt{n} \cdot \left(1 - \frac{2m}{n}\right)^{\sqrt{n}} \leq -K_M \log n,$$

i restringint que només puguem agafar com a màxim $m < n/2$ mostres, cosa que implica que $\log\left(1 - \frac{2m}{n}\right) \approx -\frac{2m}{n}$, tenim que

$$m \geq C \cdot K_M \cdot \sqrt{n} \cdot \log n.$$

Per tant, per a $f = \Delta_{\sqrt{n}}$, qualsevol algorisme necessitaria un nombre de mostres $m \sim K_M \cdot \sqrt{n} \cdot \log n$.

D'exemples per a senyals menys disperses també n'existeixen i poden construir-se a partir de l'anterior. Per tant, podem concloure que existeix un límit fonamental per al Teorema 5.1.5, i per tant no es pot recuperar qualsevol senyal a partir d'un nombre significativament petit d'observacions.

5.3 Relació amb els principis d'incertesa

Des d'un cert punt de vista, els resultats presentats en els capítols 4 i 5 estan connectats amb els *principis d'incertesa*, que diuen que és difícil localitzar la senyal $f \in \mathbb{C}^n$ a la vegada en temps i freqüència. Així, els arguments clàssics mostren que f és l'únic minimitzador de (5.1) si, i només si,

$$\sum_{t \in \mathbb{Z}_n} |f(t) + h(t)| > \sum_{t \in \mathbb{Z}_n} |f(t)|, \quad \forall h(t) \neq 0, \hat{h}|_{\Omega} = 0.$$

Signi $T = \text{supp } f$ i apliquem la desigualtat triangular. Obtenim

$$\sum_{t \in \mathbb{Z}_n} |f(t) + h(t)| = \sum_T |f(t) + h(t)| + \sum_{T^c} |h(t)| \geq \sum_T (|f(t)| - |h(t)|) + \sum_{T^c} |h(t)|.$$

Llavors, una condició suficient per establir que f és la nostra única solució seria provant que

$$\sum_T |h(t)| < \sum_{T^c} |h(t)|, \quad \forall h(t) \neq 0, \hat{h}|_{\Omega} = 0,$$

o de manera equivalent, $\sum_T |h(t)| < \frac{1}{2} \|h(t)\|_{\ell_1}$. Ara doncs, la connexió amb els principis d'incertesa és explícita; f és l'únic minimitzador és impossible concentrar la meitat de la norma ℓ_1 de la senyal a la qual li falten component de la freqüència en Ω en un conjunt T “petit”. Per exemple, Donoho [Don89], garanteix la reconstrucció exacta si

$$2|T|(n - \Omega) < n.$$

Prenent $|\Omega| < n/2$, llavors la condició diu que $|T|$ ha de ser zero, cosa que està molt lluny de ser el resultat que hem demostrat en el Teorema 5.1.5.

Refinant aquests principis d'incertesa, Donoho també demostra resultats de recuperació molt més forts. Per exemple, donada una senyal amb $|T|$ puntes repartides de qualsevol manera en el domini de temps poden ser recuperades a partir de $C \cdot |T|$ observacions de baix nivell. El Teorema 5.1.5 és distint, ja que s'aplica a *totes* les senyals que tenen el suport d'una certa mida, i no hi ha dependència amb el conjunt Ω elegit (quasi qualsevol Ω suficientment gran funcionarà). El preu per aquesta potència addicional és que requerim un factor $\log n$ més d'observacions.

Capítol 6

Recuperació determinista de senyals

Els resultats obtinguts en el capítol 5 ens ajuden a fixar un nombre mínim m de mostres pel qual es compleix la reconstrucció òptima de la senyal. Tot i així, el mètode utilitzat no és determinista, ja que implica una probabilitat de recuperació caracteritzada pel paràmetre M . En aquest capítol tornarem a resultats de caire més analític, i no tan probabilístic, demostrant que la recuperació és possible imposant unes condicions un poc més estrictes sobre les constants d'isometria restringida associades al sistema.

6.1 Propietat de reconstrucció exacta

Com veurem en la secció 6.3, amb la notació de la Definició 3.3.2, c serà l'únic minimitzador del problema (5.1) si F_T té rang màxim i existeix un vector w tal que

$$(i) \quad \langle w, v_j \rangle_H = \text{sgn}(c_j) \quad \forall j \in T,$$

$$(ii) \quad |\langle w, v_j \rangle_H| < 1 \quad \forall j \notin T,$$

on $\text{sgn}(c_j)$ és el signe de c_j o zero en el cas de que $c_j = 0$.

Definició 6.1.1. Les dues condicions anteriors ens diuen que un problema dual específic és resoluble. En conjunt, són la *Propietat de reconstrucció exacta*.

Observació 6.1.2. La propietat de reconstrucció exacta prové de les hipòtesis del Lema 5.1.1, a partir del qual Candès i Tao [Can04] fan un replantejament determinista.

Per $|T| \leq S$, amb S complint les hipòtesis del resultat central d'aquest capítol, el Teorema 6.3.1, F_T tindrà rang màxim, ja que $\delta_S < 1$ i, llavors, sols restarà construir el vector dual w ; és l'objectiu del que resta de capítol.

6.2 Propietats de reconstrucció dispersa dual

Primerament examinarem la propietat de reconstrucció dispersa per a coeficients $\langle w, v_j \rangle_H$, petits en norma ℓ_2 , per a j en T .

Lema 6.2.1 (Propietat de reconstrucció dispersa dual, versió en ℓ_2). *Siguin $S, S' \geq 1$ i tal que $\delta_S < 1$, i sigui c un vector real amb suport en $T \subset J$ tal que $|T| \leq S$. Llavors, existeix un vector $w \in H$ tal que $\langle w, v_j \rangle_H = c_j$ per a tot $j \in T$. A més, existeix un conjunt "excepcional" $E \subset J$ disjunt amb T , de cardinal fitat per S' i amb les propietats*

$$(i) \quad |\langle w, v_j \rangle_H| \leq \frac{\theta_{S,S'}}{1-\delta_S\sqrt{S'}} \|c\| \text{ per a tot } j \in T \cup E,$$

$$(ii) \quad \left(\sum_{j \in E} |\langle w, v_j \rangle_H|^2 \right)^{1/2} \leq \frac{\theta_S}{1-\delta_S} \|c\|.$$

També es dona que $\|w\|_H \leq K \cdot \|c\|$, per alguna constant $K > 0$ sols dependent de δ_S .

Demostració. Considerem ara $F_T : \ell_2(T) \rightarrow H$, que és la transformació lineal $F_T c_T := \sum_{j \in T} c_j v_j$, on $c_T := (c_j)_{j \in T}$, i sigui F_T^* la transformació adjunta

$$F_T^* w := (\langle w, v_j \rangle_H)_{j \in T}.$$

Notem que utilitzem el subíndex T en c_T per emfatitzar que l'entrada és un vector $|T|$ -dimensional.

Per la definició de les constants d'isometria restringida definides en (3.3),

$$1 - \delta_S \leq \lambda_{\min}(F_T^* F_T) \leq \lambda_{\max}(F_T^* F_T) \leq 1 + \delta_S,$$

on λ_{\min} i λ_{\max} són el mínim i màxim valor propi de l'operador definit positiu $F_T^*F_T$. En particular, com que $\delta_{|T|} < 1$, veim que $F_T^*F_T$ és invertible amb norma

$$\|(F_T^*F_T)^{-1}\| \leq \frac{1}{1 - \delta_S}. \quad (6.1)$$

A més, noti's que $\|F_T(F_T^*F_T)^{-1}\| \leq \sqrt{1 + \delta_S}/(1 - \delta_S)$ i sigui $w \in H$ el vector

$$w := F_T(F_T^*F_T)^{-1}c_T;$$

pel que queda clar doncs que $F_T^*w = c_T$, és a dir $\langle w, v_j \rangle_H = c_j$ per a tot $j \in T$. A part, $\|w\| \leq K \cdot \|c_T\|$ amb $K = \sqrt{1 + \delta_S}/(1 - \delta_S)$. Finalment, si T' és qualsevol conjunt de J que no intersequi T amb $|T'| \leq |S'|$ i $d_{T'} = (d_j)_{j \in T'}$ és qualsevol successió de nombres reals llavors, per (3.4) i (6.1), es dóna

$$\begin{aligned} |\langle F_{T'}^*w, d_{T'} \rangle_{\ell_2(T')}| &= |\langle w, F_{T'} d_{T'} \rangle_{\ell_2(T')}| = \left| \left\langle \sum_{j \in T} ((F_T^*F_T)^{-1}c_T)v_j, \sum_{j \in T'} d_j v_j \right\rangle_H \right| \\ &\leq \theta_{S,S'} \cdot \|(F_T^*F_T)^{-1}c_T\| \cdot \|d_{T'}\| \\ &\leq \frac{\theta_{S,S'}}{1 - \delta_S} \|c_T\| \cdot \|d_{T'}\|; \end{aligned}$$

com que $d_{T'}$ era arbitrari, llavors tenim per dualitat que

$$\|(F_{T'}^*w)\|_{\ell_2(T')} \leq \frac{\theta_{S,S'}}{1 - \delta_S}.$$

En altres paraules,

$$\left(\sum_{j \in T'} |\langle w, v_j \rangle|^2 \right)^{1/2} \leq \frac{\theta_{S,S'}}{1 - \delta_S} \|c_T\|, \text{ per a qualsevol } T' \subset J \setminus T \text{ i } |T'| \leq S'. \quad (6.2)$$

En particular, definim el conjunt

$$E := \left\{ j \in J \setminus T : |\langle w, v_j \rangle| > \frac{\theta_{S,S'}}{(1 - \delta_S)\sqrt{S'}} \cdot \|c_T\| \right\},$$

llavors $|E|$ ha de complir $|E| \leq S'$, ja que altrament contradiria (6.2) prenent un subconjunt T' de E de cardinal S' . Llavors, el resultat es compleix. \square

Ara en derivarem una solució millorada, amb un major control sobre la norma de $|\langle w, v_j \rangle|$ fora de T .

Lema 6.2.2 (Propietat de reconstrucció dispersa dual, versió en ℓ_∞). *Sigui $S \leq 1$ tal que $\delta_S + \theta_{S,2S} < 1$, i sigui c un vector real amb suport en $T \subset J$ que compleix $|T| \leq S$. Llavors, existeix un vector $w \in H$ tal que $\langle w, v_j \rangle_H = c_j$ per a tot $j \in T$. A més, w compleix*

$$|\langle w, v_j \rangle| \leq \frac{\theta_S}{(1 - \delta_S - \theta_{S,2S})\sqrt{S}} \cdot \|c\|, \text{ per a tot } j \notin T. \quad (6.3)$$

Demostració. Hem de normalitzar $\sum_{j \in T} |c_j|^2 = \sqrt{S}$. Escrivim $T_0 := T$. Emprant el Lema 6.2.1, podem trobar un vector $w_1 \in H$ i un conjunt $T_1 \in J$ tal que

$$\begin{aligned} T_0 \cap T_1 &= \emptyset, \\ |T_1| &\leq S, \\ \langle w_1, v_j \rangle_H &= c_j \text{ per a tot } j \in T_0, \\ |\langle w_1, v_j \rangle_H| &\leq \frac{\theta_{S,S'}}{1 - \delta_S} \text{ per a tot } j \notin T_0 \cup T_1, \\ \left(\sum_{j \in T_1} |\langle w_1, v_j \rangle_H|^2 \right)^{1/2} &\leq \frac{\theta_S}{1 - \delta_S} \sqrt{S}, \\ \|w_1\|_H &\leq K. \end{aligned}$$

Aplicant el Lema 6.2.1 recursivament obtenim una successió de vectors $w_{n+1} \in H$ i de conjunts $T_{n+1} \subset J$ per a tot $n \geq 1$ amb les propietats

$$\begin{aligned} T_n \cap (T_0 \cup T_{n+1}) &= \emptyset, \\ |T_{n+1}| &\leq S, \\ \langle w_{n+1}, v_j \rangle_H &= \langle w_n, v_j \rangle_H \text{ per a tot } j \in T_n, \\ |\langle w_{n+1}, v_j \rangle_H| &\leq \frac{\theta_S}{1 - \delta_S} \left(\frac{\theta_{S,2S}}{1 - \delta_S} \right)^n \text{ per a tot } j \notin T_0 \cup T_n \cup T_{n+1}, \\ \left(\sum_{j \in T_1} |\langle w_{n+1}, v_j \rangle_H|^2 \right)^{1/2} &\leq \frac{\theta_S}{1 - \delta_S} \left(\frac{\theta_{S,2S}}{1 - \delta_S} \right)^n \sqrt{S}, \\ \|w_{n+1}\|_H &\leq \left(\frac{\theta_S}{1 - \delta_S} \right)^{n-1} K. \end{aligned}$$

Per hipòtesis, tenim que $\frac{\theta_{S,2S}}{1 - \delta_S} \leq 1$. Llavors, si definim

$$w := \sum_{n=1}^{\infty} (-1)^{n-1} w_n,$$

la sèrie és absolutament convergent i, a més, el vector w està ben definit en H . Ara hem d'estudiar els coeficients

$$\langle w, v_j \rangle_H := \sum_{n=1}^{\infty} (-1)^{n-1} \langle w_n, v_j \rangle_H, \quad (6.4)$$

per a $j \in J$.

Considerem primer $j \in T_0$. Per construcció tenim que $\langle w_1, v_j \rangle_H = c_j$ i que $\langle w_n, v_j \rangle_H = 0$ per a tot $n \geq 2$, i llavors

$$\langle w, v_j \rangle_H = c_j, \quad \text{per a tot } j \in T_0.$$

En segon lloc, si fixem $j \notin T_0$ i sigui $I_j := \{n \geq 1 : j \in T_n\}$. Com que T_n i T_{n+1} són disjunts, tenim que els enters de I_j estan separats en almenys dues parts. Ara, si $n \in I_j$, per definició, $j \in T_n$ i llavors

$$\langle w_{n+1}, v_j \rangle_H = \langle w_n, v_j \rangle_H.$$

En altres paraules, els termes n i $n+1$ de (6.4) es cancel·len un amb l'altre. Per tant,

$$\langle w, v_j \rangle_H = \sum_{n \geq 1; n, n-1 \notin I_j} (-1)^{n-1} \langle w_n, v_j \rangle_H.$$

Per altra banda, si $n, n-1 \notin I_j$, i $n \neq 0$, llavors $j \notin T_n \cap T_{n-1}$ i

$$\langle w_n, v_j \rangle_H = \frac{\theta_{S,S}}{1 - \delta_S} \left(\frac{\theta_{S,2S}}{1 - \delta_S} \right)^{n-1},$$

la qual, amb la desigualtat triangular i la fórmula de la sèrie geomètrica, dóna

$$\left| \sum_{n \geq 1; n, n-1 \notin I_j} (-1)^{n-1} \langle w_n, v_j \rangle_H \right| \leq \frac{\theta_{S,S}}{1 - \delta_S - \theta_{S,2S}}.$$

Finalment,

$$|\langle w, v_j \rangle_H - 1_{0 \in I_j} \langle w_0, v_j \rangle_H| \leq \frac{\theta_{S,S}}{1 - \delta_S - \theta_{S,2S}},$$

i com que $|T| \leq S$, el resultat es compleix. \square

Observació 6.2.3. En efecte, el Lema 6.2.2 resol el problema de la recuperació dual. De fet, el resultat que desenvolupam afirma que hom pot trobar un vector $w \in H$ que satisfaci les propietats (i) i (ii) que en conjunt formen la propietat

de reconstrucció exacta. Per veure que (ii) es satisfà, fixem-nos que $\|\operatorname{sgn}(c)\| = \sqrt{|T|} \leq \sqrt{S}$ i, per tant, (6.3) ens dóna per a tot $j \in T$

$$|\langle w, v_j \rangle_H - 1_{0 \in I_j} \langle w_0, v_j \rangle_H| \leq \frac{\theta_{S,S}}{1 - \delta_S - \theta_{S,2S}},$$

ja que $\delta_S + \theta_{S,S} + \theta_{S,2S} < 1$.

6.3 Reconstrucció determinista

Teorema 6.3.1. *Suposem que $S \geq 1$ és tal que*

$$\delta_S + \theta_S + \theta_{S,2S} < 1, \quad (6.5)$$

i sigui c un vector real amb suport a $T \subset J$ i tal que $|T| \leq S$. Considerem $f := Fc$. Llavors, c és l'únic minimitzador de

$$\min \|d\|_{\ell_1} \quad Fd = f. \quad (6.6)$$

Demostració. Observem primer que els raonaments convexos estàndard ens asseguren l'existència d'almenys un minimitzador $d = (d_j)_{j \in J}$ del problema (6.6). Hem de provar que $d = c$. Com que c satisfà les restriccions d'aquest problema, d satisfà

$$\|d\|_{\ell_1} \leq \|c\|_{\ell_1} = \sum_{j \in T} |c_j|.$$

Ara, prenem w satisfent la propietat de reconstrucció exacta, per la qual cosa cal remetre'ns a la Observació 6.2.3. Emprant el fet que el producte $\langle w, v_j \rangle$ és igual al signe de c en T i té valor absolut estrictament menor que 1 en el complement,

llavors calculam

$$\begin{aligned}
\|d\|_{\ell_1} &= \sum_{j \in T} |c_j + (d_j - c_j)| + \sum_{j \notin T} |d_j| \\
&\geq \sum_{j \in T} \operatorname{sgn}(c_j)(c_j + (d_j - c_j)) + \sum_{j \notin T} d_j \langle w, v_j \rangle_H \\
&= \sum_{j \in T} |c_j| + \sum_{j \in T} (d_j - c_j) \langle w, v_j \rangle_H + \sum_{j \notin T} d_j \langle w, v_j \rangle_H \\
&= \sum_{j \in T} |c_j| + \left\langle w, \sum_{j \in J} d_j v_j - \sum_{j \in T} c_j \right\rangle_H \\
&= \sum_{j \in T} |c_j| + \langle w, f - f \rangle_H \\
&= \sum_{j \in T} |c_j|.
\end{aligned}$$

Comparant els dos resultats deduïts, tenim que les desigualtats han de ser igualtats. Com que $|\langle w, v_j \rangle_H|$ és estrictament menor que 1 per a tot $j \notin T$, això força que $d_j = 0$ per a tot $j \notin T$. Llavors,

$$\sum_{j \in T} (d_j - c_j) v_j = f - f = 0.$$

Aplicant (3.3) concloem que $c_j = d_j$ per a tot $j \in T$. Per tant, $d = c$ com volíem veure, donant per demostrat el teorema. \square

Observació 6.3.2. Notis que utilitzant el Lema 3.3.4 es té que (6.5) implica $\delta_{2S} < 1$, i ho implica també $\delta_S + \delta_{2S} + \delta_{3S} < 1/4$. Llavors, la condició (6.5) és tres vegades més estricta que la condició que requereix el Lema 3.3.5.

6.4 Matrius aleatòries de Gauss

Una pregunta important ara és trobar matrius que tinguin bones constants d'isometria restringida, és a dir, tal que (6.5) es compleixi per a valors relativament grans de S . D'aquesta manera, aquestes matrius toleraran una fracció bastant gran d'error mentre mantenen la informació precisa per tal d'aconseguir la reconstrucció exacta mitjançant la programació lineal.

Candès i Tao [Can05], demostren que les matrius aleatòries de Gauss són una classe de matrius per a la qual hom pot resoldre un sistema d'equacions lineals

indeterminat minimitzant la norma ℓ_1 , sempre que el vector introduït tingui menys de $\rho \cdot m$ entrades distintes a zero, per un $\rho > 0$. A més, obtenen valors numèrics explícits sobre les condicions per obtenir la recuperació exacta.

Considerem una matriu F de p per m de tal manera que les seves entrades siguin independents i idènticament distribuïdes seguint una distribució gaussiana amb mitja zero i variança $1/p$. Sigui també T un subconjunt de les columnes de F . Volem estudiar els valors extrems dels valors propis de $F_T^* F_T$. En els seus treballs, Pastur [Mar67], Geman [Gem80] i Silverstein, [Sil85] provaren que

$$\begin{aligned}\lambda_{\min}(F_T^* F_T) &\rightarrow (1 - \sqrt{\gamma})^2 \text{ q.s.}, \\ \lambda_{\max}(F_T^* F_T) &\rightarrow (1 + \sqrt{\gamma})^2 \text{ q.s.},\end{aligned}$$

en el límit quan p i $|T| \rightarrow \infty$ amb

$$|T|/p \rightarrow \gamma \leq 1.$$

Amb altres paraules, que quan p creix molt la constant d'isometria $\delta(F_T)$ per a un T fixat es comporta com

$$1 - \delta(F_T) \leq \lambda_{\min}(F_T^* F_T) \leq \lambda_{\max}(F_T^* F_T) \leq 1 + \delta(F_T), \quad \delta(F_T) \approx 2\sqrt{|T|/p} + |T|/p.$$

Les constants d'isometria restringida s'han de satisfer per a tots els conjunts T de cardinal menor o igual a S , i hem de fer ús de desigualtats de concentració per tal de trobar una fita uniforme. Noti's que donat un $T' \subset T$, tenim

$$\lambda_{\min}(F_T^* F_T) \leq \lambda_{\min}(F_{T'}^* F_{T'}) \leq \lambda_{\max}(F_{T'}^* F_{T'}) \leq \lambda_{\max}(F_T^* F_T),$$

i per tant, l'atenció ha de residir en les matrius de dimensió S . Però hi ha resultats de grans desviacions sobre els valors singulars de F_T [Sza91]. Per exemple, sigui $\sigma_{\max}(F_T)$ (respectivament σ_{\min}) el valor singular més gran de F_T tal que $\sigma_{\max}^2(F_T) = \lambda_{\max}(F_T^* F_T)$ (respectivament $\sigma_{\min}^2(F_T) = \lambda_{\min}(F_T^* F_T)$), Ledoux [Led01] aplica la desigualtat de concentració per a mesures gaussianes, i per a cada $t > 0$ fixat, obté les fites

$$\mathbb{P}\left(\sigma_{\max}(F_T) > 1 + \sqrt{|T|/p} + O(1) + t\right) \leq e^{-pt^2/2}, \quad (6.7)$$

$$\mathbb{P}\left(\sigma_{\max}(F_T) < 1 - \sqrt{|T|/p} + O(1) - t\right) \leq e^{-pt^2/2}. \quad (6.8)$$

Aquí, $O(1)$ és un terme petit que tendeix a zero així com $p \rightarrow \infty$ i que pot ser calculat explícitament, [ElK04]. Per exemple, la darrera referència mostra que hom pot seleccionar $O(1)$ en (6.7) com $\frac{1}{2p^{1/3}} \cdot \gamma^{1/6}(1 + \sqrt{\gamma})^{2/3}$.

Lema 6.4.1. *Sigui $r = S/m$ i sigui*

$$f(r) := \sqrt{m/p} \left(\sqrt{r} + \sqrt{2H(r)} \right),$$

on H és la funció d'entropia $H(q) := -q \log q - (1 - q) \log(1 - q)$ definida per a $0 < q < 1$. Per a cada $\varepsilon > 0$, la constant d'isometria restringida δ_S d'una matriu gaussiana F de dimensió $p \times m$ compleix

$$\mathbb{P} \left(1 + \delta_S > [1 + (1 + \varepsilon)f(r)]^2 \right) \leq 2e^{-mH(r)\varepsilon/2}.$$

Demostració. Com hem dit abans, ens restringirem la nostra atenció als conjunts T tal que $|T| = S$, denotem-ho $\mathbf{T}_S = \{T : |T| = S\}$. Denotem per η_p el terme $O(1)$ que apareix tant en (6.7) com en (6.8). Per abreujar, escriurem $\lambda_{\max} = \lambda_{\max}(F_T^* F_T)$, i notem que

$$\begin{aligned} \mathbb{P} \left(\sup_{\mathbf{T}_S} \lambda_{\max} > (1 + \sqrt{S/p} + \eta_p + t)^2 \right) &\leq |\mathbf{T}_S| \mathbb{P} \left(\lambda_{\max} > (1 + \sqrt{S/p} + \eta_p + t)^2 \right) \\ &\leq \binom{m}{S} e^{-pt^2/2}. \end{aligned}$$

De les aproximacions de Stirling $\log m! = m \log m - m + O(\log m)$ tenim la fórmula

$$\log \binom{m}{S} = mH(r) + O(\log m),$$

la qual ens dóna

$$\mathbb{P} \left(\sup_{\mathbf{T}_S} \lambda_{\max} > (1 + \sqrt{S/p} + \eta_p + t)^2 \right) \leq e^{mH(r)} e^{O(\log m)} e^{-pt^2/2}.$$

El mateix raonament aplicat al valor propi menor ens dóna

$$\mathbb{P} \left(\inf_{\mathbf{T}_S} \lambda_{\min} < (1 - \sqrt{S/p} - \eta_p - t)^2 \right) \leq e^{mH(r)} e^{O(\log m)} e^{-pt^2/2}.$$

Fixem $\eta_p + t = (1 + \varepsilon)\sqrt{m/p}\sqrt{2H(r)}$, i suposem que m i p són prou grans per complir $\eta_p \leq \varepsilon/2\sqrt{m/p}\sqrt{2H(r)}$. Llavors,

$$\mathbb{P} \left(\sup_{\mathbf{T}_S} \lambda_{\max} > (1 + \sqrt{S/p} + \varepsilon\sqrt{m/p}\sqrt{2H(r)})^2 \right) \leq e^{-mH(r)\varepsilon/2},$$

on hem emprat el fet que el terme $O(\log m)$ és més petit que $m\varepsilon H(r)/2$ per a m prou gran. Tenim la mateixa fita per al valor propi mínim i, per tant, es satisfà el resultat. \square

Ignorant les ε , el Lema 6.4.1 afirma que amb una probabilitat molt gran es compleix

$$\delta_S < -1 + [1 + f(r)]^2.$$

Una conclusió similar es té per δ_{2S} i per δ_{3S} i, per tant, tenim que

$$\delta_S + \delta_{2S} + \delta_{3S} < \rho_{p/m}(r), \quad \rho_{p/m}(r) = \sum_j = 1^3 - 1 + [1 + f(jr)]^2,$$

amb una probabilitat de nou molt alta. Concloent, el Lema 3.3.4 mostra que la hipòtesi del Teorema 6.3.1 es compleix sempre que la ratio $r = S/m$ sigui prou petita com perquè $\rho_{p/m} < 1$. En altres paraules, que el límit per a mostrejos grans, S/m s'ha d'agafar com a qualsevol valor que satisfaci $\rho_{p/m}(S/m) < 1$, el qual, gràcies a resultats de [Can05], es pot emprar per a prendre fites¹ numèriques útils per trobar matrius aleatòries de Gauss per a les quals obtinguem la recuperació exacta.

¹Bàsicament, si deim $r := S/m$, obtindrem un $r^*(p, m)$ màxim, de tal manera que per assegurar la reconstrucció s'haurà de complir $r < r^*(p, m)$.

Capítol 7

Exemples

Un cop presentada la teoria, ens resultarà útil comprovar els resultats obtinguts, desenvolupant prèviament i de manera esquemàtica el procés de reconstrucció a seguir. Com hem mostrat en la secció 1.4.3, hi ha moltes variants del problema.

Per a primer exemple prendrem el tipus de senyal i problema d'optimització pel qual hem realitzat l'estudi. Per tant, resoldrem un programa lineal (LP) emprant un mètode genèric primal-dual de seguiment de camins.

En el segon, que serà més visual, prendrem senyals bidimensionals minimitzant la variació total definida a (1.4). En aquest cas, necessitarem un algorisme log-barrera, ja que haurem de treballar amb un programa cònic de segon ordre (SOCP).

Notem que hem pres de [Can04] algunes de les rutines creades per Romberg per a realitzar els següents exemples. Unes quantes els empram tal i com ell les dissenyà, d'altres les hem modificades, i unes terceres són de creació pròpia.

7.1 ℓ_1 amb restriccions d'igualtat

Suposem que volem resoldre el problema (5.1), on la senyal original és un vector de N components entre les quals se n'han seleccionat T que prenen valors equiprobablement $+1$ i -1 .

Per prendre les m mostres, construïm una matriu aleatòria A amb una distribució uniforme i la ortogonalitzem. Prenem el conjunt de mostres y com el producte de A per la senyal x i el punt inicial com el producte $A^T y$, fent notar que $AA^T = Id$.

7.1.1 Algorisme primal-dual

Per tal de resoldre el sistema d'optimització, hem de recórrer a [Boy04], on Boyd i Vandenberghe desenvolupen un algorisme primal-dual per a programació lineal relativament senzill.

La forma estàndard del problema lineal és

$$\begin{aligned} \min_z \langle c_0, z \rangle \text{ subjecte a } A_0 z = b, \\ f_i(z) \leq 0, \end{aligned}$$

on el vector cercat $z \in \mathbb{R}^N$, $b \in \mathbb{R}^K$, i A_0 és una matriu de dimensió $K \times N$, i on cada un dels f_i , $i = 1, \dots, m$, és un funcional lineal tal que $f_i(z) = \langle c_i, z \rangle + d_i$, per algun $c_i \in \mathbb{R}^N$, $d_i \in \mathbb{R}$.

Al punt òptim z^* , existiran vectors duals $\nu^* \in \mathbb{R}^K$, $\lambda^* \in \mathbb{R}^m$, $\lambda^* \geq 0$ tal que les condicions de Karush-Kuhn-Tucker es satisfan:

$$\begin{aligned} c_0 + A_0^T \nu^* + \sum_i \lambda_i^* c_i &= \mathbf{0}, \\ \lambda_i^* f_i(z^*) &= 0, \quad i = 1, \dots, m, \\ A_0 z^* &= b, \\ f_i(z^*) &\leq 0, \quad i = 1, \dots, m. \end{aligned} \tag{KKT}$$

En resum, l'algorisme primal-dual troba l'òptim z^* (així com els vectors duals ν^* i λ^*) resolent el sistema d'equacions no lineals previ. La solució és el mètode de Newton clàssic: en un punt interior (z^k, ν^k, λ^k) , és a dir, per $f_i(z^k) < 0$ i $\lambda^k > 0$, el sistema es linealitzat i resolt. Durant el seu desenvolupament pot ser haurem de modificar el nou punt $(z^{k+1}, \nu^{k+1}, \lambda^{k+1})$ per a que es mantengui en l'interior.

En la pràctica, relaxarem la hipòtesi complementària de laxitud $\lambda_i f_i = 0$ canviant-la per

$$\lambda_i^k f_i(z^k) = -1/\tau^k, \tag{7.1}$$

on anirem augmentat controladament el valor del paràmetre τ^k a través de les iteracions de Newton.

Això provoca el biaix de la solució del sistema d'equacions linealitzat cap a l'interior, permetent l'existència d'un "camí central", ben definit i diferenciable, des d'un punt interior cap a la solució a la frontera.

Els residus primal, central i dual mesuren com d'aprop està el punt (z, ν, λ) de

satisfir les hipòtesis (KKT) amb (7.1) en el lloc de la hipòtesis de laxitud:

$$\begin{aligned} r_{\text{dual}} &= c_0 + A_0^T \nu + \sum_i \lambda_i c_i, \\ r_{\text{cent}} &= -\Lambda f - (1/\tau)Id, \\ r_{\text{pri}} &= A_0 z - b, \end{aligned}$$

on Λ és una matriu diagonal amb $(\Lambda)_{ii} = \lambda_i$, i $f = (f_1(z) \dots f_m(z))^T$.

Des d'un punt (z, ν, λ) , volem trobar el pas $(\Delta z, \Delta \nu, \Delta \lambda)$ tal que

$$r_\tau(z + \Delta z, \nu + \Delta \nu, \lambda + \Delta \lambda) = 0. \quad (7.2)$$

Linealitzant (7.2) amb l'expansió de Taylor al voltant de (z, ν, λ) ,

$$r_\tau(z + \Delta z, \nu + \Delta \nu, \lambda + \Delta \lambda) \approx r_\tau(z, \nu, \lambda) + J_{r_\tau}(z, \nu, \lambda) \begin{pmatrix} \Delta z \\ \Delta \nu \\ \Delta \lambda \end{pmatrix},$$

on $J_{r_\tau}(z, \nu, \lambda)$ és la Jacobiana de r_τ , tenim el sistema

$$\begin{pmatrix} \mathbf{0} & A_0^T & C^T \\ -\Lambda C & \mathbf{0} & -F \\ A_0 & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta \nu \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} c_0 + A_0^T \nu + \sum_i \lambda_i c_i \\ -\Lambda f - (1/\tau)Id \\ b - A_0 z \end{pmatrix},$$

on la matriu C , de dimensió $m \times N$, té les c_i^T com a files, i F és diagonal amb $(F)_{ii} = f_i(z)$. Podem eliminar $\Delta \lambda$, emprant

$$\Delta \lambda = -\Lambda F^{-1} X \Delta z - \lambda - (1/\tau) f^{-1}, \quad (7.3)$$

deixant el sistema com

$$\begin{pmatrix} -C^T F^{-1} \Lambda C & A_0^T \\ A_0 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} -c_0 + (1/\tau) C^T f^{-1} - A_0^T \nu \\ -\Lambda f - (1/\tau) Id \\ b - A_0 z \end{pmatrix}. \quad (7.4)$$

Amb $(\Delta z, \Delta \nu, \Delta \lambda)$ ja tenim la direcció del pas. Per a triar la llargada $0 < s \leq 1$, miram si es satisfan els següents dos criteris:

1. $z + s\Delta z$ i $\lambda + s\Delta \lambda$ estan en l'interior, és a dir, $f_i(z + s\Delta z) < 0$, $\lambda_i > 0$, per a tot i .
2. La norma dels residus ha minvat suficient:

$$\|r_\tau(z + s\Delta z, \nu + s\Delta \nu, \lambda + s\Delta \lambda)\|_2 \leq (1 - \alpha s) \cdot \|r_\tau(z, \nu, \lambda)\|_2,$$

on α es un paràmetre, pel qual $\alpha = 0.01$ és un bon valor.

Com que f_i són funcionals lineals, el punt 1 és fàcilment deduïble. Triarem el pas més gran que ens mantengui a l'interior. Sigui

$$\mathcal{I}_f^+ = \{i : \langle c_i, \Delta z \rangle > 0\}, \quad \mathcal{I}_f^- = \{i : \Delta z < 0\},$$

i diguem

$$s_{\max} = 0.99 \cdot \{1, \{-f_i(z)/\langle c_i, \Delta z \rangle, i \in \mathcal{I}_f^+\}, \{-\lambda_i/\Delta \lambda_i, i \in \mathcal{I}_f^-\}\}.$$

Llavors, començant en $s = s_{\max}$, comprovem si el punt 2 es satisfà; si no, posem $s' = \beta \cdot s$ i ho intentem de nou. Un bon valor per β és $1/2$.

Quan r_{dual} i r_{pri} són petits, el “substitut del forat de dualitat”, $\eta = -f^T \lambda$, és una aproximació de com d'aprop està un cert (z, ν, λ) de ser òptim, és a dir, de ser tal que $\langle c_0, z \rangle - \langle c_0, z^* \rangle \approx \eta$. L'algorisme primal-dual va realitzant passes de Newton com hem descrit fins que η ha tornat més petit que una certa tolerància preestablerta.

Quasi tot el pes de l'algorisme recau en resoldre (7.4). Quan la matriu $-C^T F^{-1} \Delta C$ és fàcilment invertible, com és el nostre cas, (7.4) pot ser reduït a un sistema d'equacions simètric i definit positiu.

Per N i K molt grans resulta impossible resoldre sistemes com (7.4). Hi ha algorismes iteratius, com el dels Gradients Conjugats descrits a [She94], que fan possible esquivar aquest problema.

7.1.2 Concreció de l'algorisme

Quan x , A i b són reals, el problema (5.1) es pot escriure com

$$\begin{aligned} \min_{x,u} \sum_i u_i \quad \text{subjecte a } & x_i - u_i \leq 0, \\ & -x_i - u_i \leq 0, \\ & Ax = b, \end{aligned}$$

el qual es pot resoldre amb l'algorisme primal-dual esmentat a la subsecció 7.1.1 concretant com agafem el pas de Newton. Considerem

$$\begin{aligned} f_{u_1;i} &:= x_i - u_i, \\ f_{u_2;i} &:= -x_i - u_i, \end{aligned}$$

amb $\lambda_{u_1;i}, \lambda_{u_2;i}$ les variables duals associades, i sigui f_{u_1} el vector $(f_{u_1;1} \dots f_{u_1;N})^T$ (i definim f_{u_1}, λ_{u_1} i λ_{u_2} de la mateixa manera). Noti's que

$$\nabla f_{u_1;i} = \begin{pmatrix} \delta_i \\ -\delta_i \end{pmatrix}, \quad \nabla f_{u_2;i} = \begin{pmatrix} -\delta_i \\ -\delta_i \end{pmatrix}, \quad \nabla^2 f_{u_1;i} = 0, \quad \nabla^2 f_{u_2;i} = 0,$$

on δ_i és el vector de la base estàndard per a la component i . Llavors, al punt $(x, u; v, \lambda_{u_1}, \lambda_{u_2})$, els residuals centrals i duals són

$$r_{\text{cent}} = \begin{pmatrix} -\Lambda_{u_1} f_{u_1} \\ -\Lambda_{u_2} f_{u_2} \end{pmatrix} - (1/\tau) Id,$$

$$r_{\text{dual}} = \begin{pmatrix} \lambda_{u_1} - \lambda_{u_2} + A^T v \\ Id - \lambda_{u_1} - \lambda_{u_2} \end{pmatrix},$$

i el pas de Newton (7.4) ve donat per

$$\begin{pmatrix} \Sigma_1 & \Sigma_2 & A^T \\ \Sigma_2 & \Sigma_1 & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \\ \Delta v \end{pmatrix} = \begin{pmatrix} \Delta w_1 \\ \Delta w_2 \\ \Delta w_3 \end{pmatrix} := \begin{pmatrix} (-1/\tau) \cdot (-f_{u_1}^{-1} + f_{u_2}^{-1}) - A^T v \\ -Id - (1/\tau) \cdot (f_{u_1}^{-1} + f_{u_2}^{-1}) \\ b - Ax \end{pmatrix},$$

amb

$$\Sigma_1 = \Lambda_{u_1} F_{u_1}^{-1} - \Lambda_{u_2} F_{u_2}^{-1}, \quad \Sigma_2 = \Lambda_{u_1} F_{u_1}^{-1} + \Lambda_{u_2} F_{u_2}^{-1}.$$

Noti's ara que les F_{\bullet} , per exemple, són matrius diagonals amb $(F_{\bullet})_{ii} = f_{\bullet;i}$, i $f_{\bullet;i}^1 = 1/f_{\bullet;i}$. Llavors, si deim

$$\Sigma_x = \Sigma_1 - \Sigma_2^2 \Sigma_2^{-1},$$

podem eliminar

$$\Delta x = \Sigma_x^{-1} (w_1 - \Sigma_2 \Sigma_1^{-1} w_2 - A^T \Delta v),$$

$$\Delta u = \Sigma_1^{-1} (w_2 - \Sigma_2 \Delta x),$$

i resoldre

$$-A \Sigma_x^{-1} A^T \Delta v = w_3 - A (\Sigma_x^{-1} w_1 - \Sigma_x^{-1} \Sigma_2 \Sigma_1^{-1} w_2).$$

Aquest sistema d'equacions és de dimensió $K \times K$ i definit positiu, pot ser resolt mitjançant el gradient conjugat. Donats $\Delta x, \Delta u, \Delta v$, calculam els canvis en les variables duals de la desigualtat com en (7.3):

$$\Delta \lambda_{u_1} = \Lambda_{u_1} F_{u_1}^{-1} (-\Delta x + \Delta u) - \delta_{u_1} - (1/\tau) f_{u_1}^{-1},$$

$$\Delta \lambda_{u_2} = \Lambda_{u_2} F_{u_2}^{-1} (\Delta x + \Delta u) - \delta_{u_2} - (1/\tau) f_{u_2}^{-1}.$$

7.1.3 Rutines

A continuació podeu veure una breu explicació de què fa cada algorisme dels utilitzats el l'exemple que ens ocupa. El codi en Matlab de cada un d'ells el trobareu a l'annex A, on també hi haurà més informació de cada una de les rutines, així com dels paràmetres que se li han d'introduir i quins retorna.

recMin.m

Donat un interval de mostres K i de pics T pot donar dos resultats. Si l'interval de T només conté un element, retorna el gràfic de l'error de recuperar una imatge consistent en T pics per als diferents valors de K . Si l'interval de T conté més d'un element, retorna, per a cada valor de K , el primer valor per al qual es recupera la imatge en F i el valor a partir del qual es recupera la imatge G . En el segon dels casos, el paràmetre *enMitja*, dóna els valors de recuperació teòrica prenent la mitja de les μ per a aplicar la fórmula.

l1eq_example.m

Donada una senyal unidimensional de longitud N consistent en T pics presos aleatòriament en l'espai i que poden valer $+1$ o -1 , un paràmetre d'error M , recupera una aproximació xp de la senyal havent pres prèviament K mostres. També retorna l'error i el valor de μ . En el cas de no incloure cap paràmetre, el programa funciona de forma interactiva i demana a l'usuari els que necessita.

l1eq_pd.m

Troba el mínim de $\min_x \|x\|_1$ tal que $Ax = b$ utilitzant un algorisme de punt interior primal-dual.

cgsolve.m

Resol un sistema simètric i definit positiu de la forma $Ax = b$ utilitzant el mètode dels gradients conjugats.

7.1.4 Resultats

En aquest exemple es crea una senyal unidimensional de llargària $N := 512$ punts, on hi afegim $T := 20$ pics que valen $+1$ o -1 . La situació i valor d'aquests es prenen aleatòriament en la rutina `l1ep_example.m`. Si llençam l'esmentada rutina sense paràmetres, per a dimensió N i paràmetre de l'error $M := 1.3$, i si introduïm el valor de pics T i mostres¹ preses $K := 86$ (un 16.8% del total de punts), llavors obtenim els gràfics que apareixen a la figura 7.1.

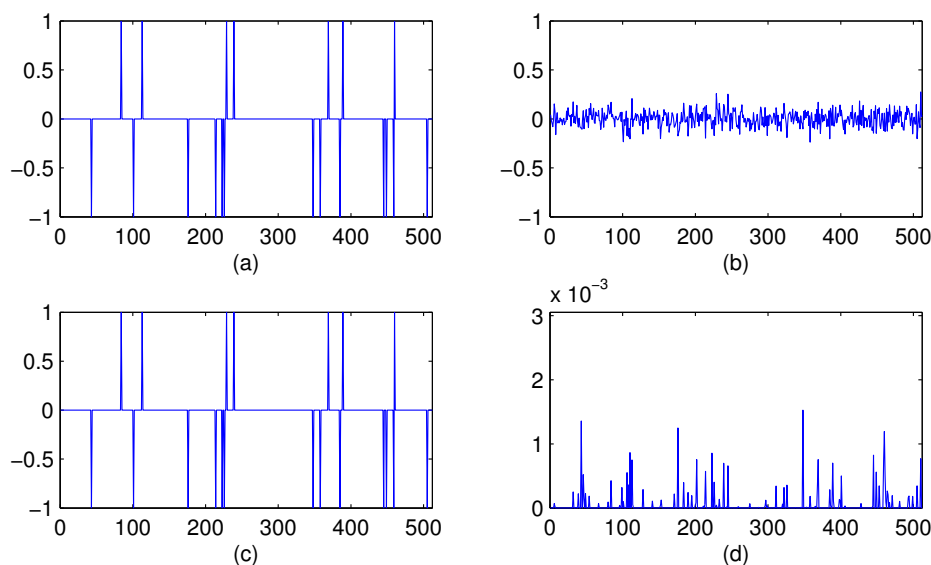


Figura 7.1: Per als paràmetres $N = 512$, $T = 20$, $K = 86$ i $M = 1.3$: (a) Senyal original. (b) Mínima energia. (c) Senyal recuperada. (d) Error absolut en la recuperació.

Primer de tot, la rutina `l1ep_example.m` comprova una condició que s'ha de satisfer per a que l'aproximació que prendrem de C_M sigui correcte. Aquesta aproximació prové de la demostració de Lema 4.4.4 i és la següent:

$$(1.5 - 0.87(M + 1)) \log N \leq -3.10.$$

¹Notem que desenvolupant la teoria empràvem la lletra m per al nombre de mostres, igual que n per a la dimensió. A partir d'aquest moment, utilitzarem la lletra K per al nombre de mostres indistintament amb m . Pel que fa a la dimensió, utilitzarem n com mesura en cada una de les dimensions i N com a nombre total d'elements de la senyal. Per tant, en dimensió 1, $n = N$.

Notem que el paràmetre M que hem pres compleix aquesta condició, ja que

$$(1.5 - 0.87(1.3 + 1)) \log 512 = -3.1254 < -3.10.$$

A la part (a) de la figura 7.1 trobem la imatge original, generada aleatòriament a partir d'una longitud donada i del nombre de pics que triem. Després de crear la imatge, el programa genera una aproximació de la fita mínima per a la qual s'espera la recuperació amb una probabilitat de $p = 1 - N^{-M} = 0.9997$. Ho fa aplicant el Teorema 5.1.5. Per al valor de μ , el programa crea una aproximació de la matriu de mostreig A prenent un 20% de N com al nombre K de mostres. Feim això perquè, la matriu de mostreig A de la qual μ depèn, la prenem ortogonalitzant una matriu presa aleatòriament seguint una distribució uniforme i, per tant, el valor de μ variarà dependent d'ella. En el nostre cas, obtenim $\tilde{\mu} = 0.1208$, i per tant la fita és

$$23(M + 1)\tilde{\mu}^2 T \log(N) = 83.6$$

En aquest punt, el programa és bastant sensible respecte al valor de μ , i pot ser que obtenguem una fita lleugerament més petita del valor mínim de recuperació, per a la qual la fita no serà vàlida. De la mateixa manera, pot ser que obtenguem un valor més gran, i per tant que la fita no sigui òptima.

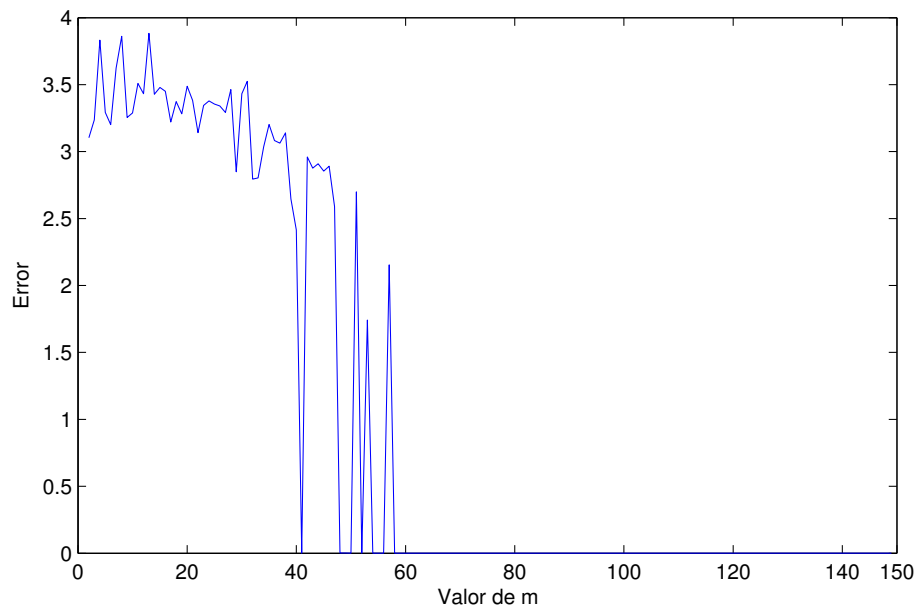


Figura 7.2: Error en la recuperació en funció del nombre m de mostres per a $T = 10$.

Si prenem, com hem dit, $K = 86$, a la part (b) de la figura 7.1 podem veure la senyal x_0 que es pren com a punt inicial per a resoldre el problema de programació lineal, també anomenada senyal de mínima energia. Si x és la senyal que volem reconstruir i A és la matriu de mostreig obtinguda, llavors les mostres preses venen donades per $y = A * x$ i el punt inicial és $x_0 = A' * y$. D'aquesta manera, per ser A ortogonal, el punt satisfà trivialment el sistema.

A l'apartat (c) de la figura 7.1 podem comprovar com la recuperació és, a simple vista, perfecta. La fita esdevé correcta i la senyal és indistingible. Obtenim aquesta aproximació mitjançant la subrutina `l1eq_pd.m`, que resol el sistema de programació lineal utilitzant un mètode de punt interiors amb un algorisme primal-dual. En l'apartat (d) de la mateixa figura hi ha l'error obtingut, que és de l'ordre de 10^{-3} . Augmentant el valor de les mostres per sobre de la fita, o augmentant el valor de M , obtindrem millors aproximacions de la senyal.

Notem que el programa retorna com a variables, la imatge reconstruïda x_p , l'error en la recuperació `ERROR`, així com `MU`, el valor de la μ recalculada amb la matriu de mostreig final A , que alhora depèn del K finalment pres.

Com hem pogut comprovar, la recuperació és possible i es pot dur a la pràctica. Un altre punt interessant és comprovar la fiabilitat del nostre resultat principal, el Teorema 5.1.5.

Primer de tot, el que volem comprovar és que, a partir d'un nombre de mostres K la senyal sempre és reconstruïble. Llavors, si això succeeix, farem una segona passa: Si busquem el primer valor de K pel qual s'ha pogut reconstruir la senyal i el mínim valor de K per al qual *sempre* és reconstrueix la senyal, llavors tendrem localitzat aproximadament quin seria el valor de la fita, i el podríem comparar amb el del Teorema 5.1.5. El les figures 7.2 i 7.3 podem trobar, respectivament per als valors de $T = 10$ i de $T = 35$, el gràfic de l'error obtingut en la recuperació en funció del nombre de mostres preses.

En el primer cas, és a dir, en la figura 7.2, podem veure com obtenim la primera recuperació sobre el valor de $K = 40$ (7.8%) i la recuperació exacta a partir d'un valor de K propera a 60 (11.7%). A més, la recuperació és de cada vegada més probable i, en el cas de no obtenir-la, el resultat és més pròxim al desitjat. En la figura 7.3, podem veure com la tendència és la mateixa, però traslladat a valors més grans de K . Ara, la primera recuperació és per a un valor de $K = 125$ (24.4%), i la recuperació segura a partir de $K = 145$ (28.3%). Aquests resultats els hem aconseguit llençant la rutina `recMin.m` amb els paràmetres següents:

- `recMin(5,150, 10, 10, 512, 1.3, [])`; (Fig. 7.2)

- `recMin(25,180, 35, 35, 512, 1.3, [])`; (Fig. 7.3)

Amb la mateixa rutina, simplement canviant els valors dels paràmetres `nt` i `nT` obtenim un altre tipus de resultats. Si els valors d'aquests dos paràmetres no són iguals, el programa cerca, per a cada valor de T dins l'interval introduït, el primer valor de K pel qual obtenim la recuperació exacta, el mínim K a partir del qual sempre obtenim recuperació juntament amb la fita teòrica donada pel Teorema 5.1.5 amb la μ presa segons la matriu de mostreig A . D'aquesta manera, veurem com es comporta la fita teòrica al costat dels valors reals obtinguts en les proves.

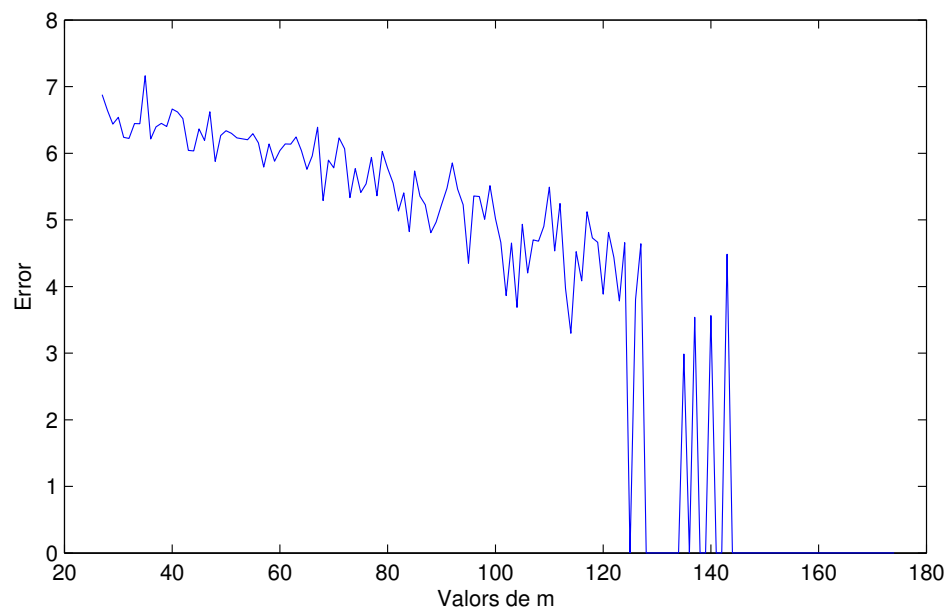


Figura 7.3: Error en la recuperació en funció del nombre m de mostres per a $T = 35$.

En la figura 7.4 podem veure la sensibilitat del mínim teòric, representat per la línia puntejada, que prové de la sensibilitat de la μ . Així com creix T també creix la fluctuació de la fita, ja que multiplica a la μ i accentua l'efecte provinent de l'aleatorietat de la matriu de mostreig, que incideix directament sobre μ . Tot i així, podem comprovar com els valors són els esperats, ja que estan al voltant dels valors de recuperació segura i de primera recuperació obtinguts empíricament.

Per tal de suavitzar l'efecte de l'aleatorietat, prenem un μ únic a l'hora de calcular la fita teòrica del Teorema 5.1.5, que ara sols dependrà de T . El μ únic el prenem

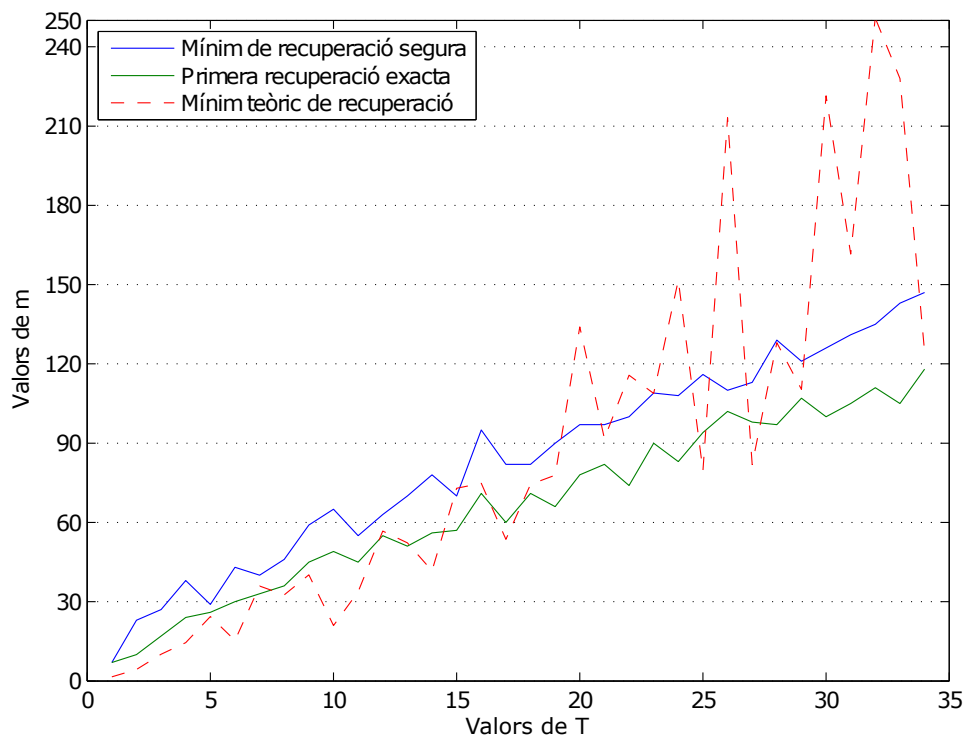


Figura 7.4: Mínims de recuperació en funció del nombre de pics i μ variable.

com la mitja aritmètica dels diferents valors de μ en l'interval donat, obtenint la gràfica de la figura 7.5. En aquesta gràfica podem observar que el valor que ens dona la fita teòrica està entre la primera recuperació i la recuperació segura. Aquest fet recalca el l'afirmació del resultat principal que presentam, que assegura una recuperació amb una certa probabilitat, en concret $p = 1 - N^{-M}$.

Notem que el aquest estudi cercava la recuperació de senyals a partir d'un nombre molt petit de mostres, això és, tal que $K \ll N$, on recordem que K són les mostres i N el nombre d'elements de la senyal. Per aquest motiu, els valors de K presos per a comprovar aquests resultats són relativament petits respecte N , ja que és el que afirmen els resultats provats. Per tant, per a valors de K pròxims a N , tot i que sí que s'espera la recuperació exacta, no s'espera que la fita per al valor de K sigui correcte.

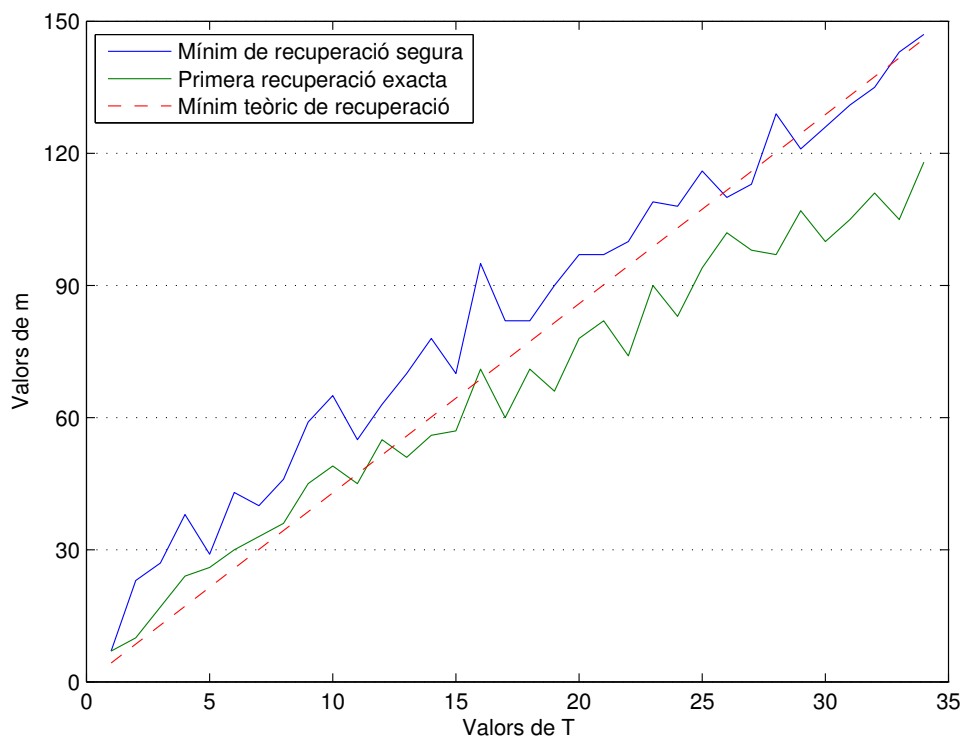


Figura 7.5: Mínims de recuperació en funció del nombre de pics i μ en mitja.

7.2 $TV(x)$ amb restriccions d'igualtat

En aquest cas, prendrem un exemple de gran dimensió. Prendrem una imatge de 256×256 i, després d'agafar-ne mostres per mitjà de dues màscares diferents (una màscara amb L línies radials i una altra amb P punts presos seguint una distribució uniforme), la recuperarem mitjançant un algorisme log-barrera. Hem d'utilitzar aquest algorisme perquè obtenim el següent problema cònic de segon orde:

$$\min TV(x) \quad \text{subjecte a} \quad Ax = b, \quad (7.5)$$

on $TV(x)$ és el definit a (1.4).

7.2.1 Algorisme log-barrera

Tot i que existeixen algorismes de tipus primal-dual per a resoldre problemes cònics de segon ordre, aquests tornen bastants més complicats que en el cas dels LP. Per tant, s'ha decidit implementar els problemes SOCP emprant el mètode genèric log-barrera descrit en [Boy04]. Aquest mètode, tot i que conceptualment és més directe que el primal-dual, també es basa en aplicar passes de Newton.

El problema (7.5) es pot escriure com

$$\begin{aligned} \min_z \langle c_0, z \rangle \text{ subjecte a } A_0 z = b, \\ f_i(z) \leq 0, i = 1, \dots, m, \end{aligned} \quad (7.6)$$

on cada un dels f_i , $i = 1, \dots, m$, és un funcional lineal tal que $f_i(z) = \langle c_i, z \rangle + d_i$, o

$$f_i(z) = \frac{1}{2}(\|A_i z\|_2^2 - (\langle c_i, z \rangle + d_i)^2),$$

on A_i són matrius, c_i vectors i d_i escalars.

El mètode bàsic log-barrera transforma (7.6) en una sèrie de programes amb restriccions lineals de la forma

$$\min_z \langle c_0, z \rangle + \frac{1}{\tau^k} \sum_i -\log(-f_i(z)) \text{ subjecte a } A_0 z = b, \quad (7.7)$$

on $\tau^k > \tau^{k-1}$. D'aquesta manera, hem incorporat les restriccions dins el funcional via una funció de penalització. El fet de que en el sumatori aparegui aquest logaritme no és arbitrari, i resulta molt important per a la convergència ràpida de (7.6) i (7.7). La funció de penalització esmentada farà que el funcional sigui infinit si les condicions que substitueix són violades o, altrament, que sigui diferenciable per tot. Així com τ^k creix, la solució z^k de (7.7) s'aproxima a la solució z^* de (7.6): es pot veure que $\langle c_0, z^k \rangle - \langle c_0, z^* \rangle < m/\tau^k$, és a dir, que estem a menys de m/τ^k del valor òptim després de la iteració k (m/τ^k s'anomena el forat dual).

La idea aquí és que cada subproblema pot ser resolt raonablement bé amb poques iteracions del mètode de Newton, particularment si podem emprar la solució z^k com a inicial per a trobar z^{k+1} .

A la iteració log-barrera k , el mètode de Newton (que torna a ser iteratiu) va formant sèries d'aproximacions quadràtiques de (7.7), minimitzant-ne cada una resolent un sistema d'equacions. Igual que en el cas anterior, a cada passa haurem

de prendre una longitud prou petita per a que romandre en l'interior. L'aproximació quadràtica del funcional

$$f_0(z) = \langle c_0, z \rangle + \frac{1}{\tau} \sum_i -\log(-f_i(z))$$

en (7.7) al voltant del punt z ve donada per

$$f_0(z + \Delta z) \approx z + \langle g_z, \Delta z \rangle + \frac{1}{2} \langle H_z \Delta z, \Delta z \rangle := q(z + \Delta z),$$

on g_z és el gradient

$$g_z = c_0 + \frac{1}{\tau} \sum_i \frac{1}{-f_i(z)} \nabla f_i(z),$$

i H_z és la matriu Hessiana

$$H_z = \frac{1}{\tau} \sum_i \frac{1}{f_i(z)^2} \nabla f_i(z) (\nabla f_i(z))^T + \frac{1}{\tau} \sum_i \frac{1}{-f_i(z)} \nabla^2 f_i(z).$$

Si z és un valor possible (en particular es compleix $A_0 z = b$), el Δz que minimitza $q(z + \Delta z)$ subjecte a $A_0 z = b$ és la solució del conjunt d'equacions lineals

$$\tau \begin{pmatrix} H_z & A_0^T \\ A_0 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta z \\ v \end{pmatrix} = -\tau g_z. \quad (7.8)$$

Noti's que el vector v , que pot ser interpretat com els multiplicadors de Lagrange en les restriccions del problema quadràtic de minimització, no és emprat directament.

Com que el problema que ens ocupa és serà simètric però indefinit, emprarem l'algorisme SYMMLQ, és a dir, un algorisme LQ per a sistemes lineals simètrics. Havent pres Δz , tenim la direcció de les passes per a Newton. El pas, de longitud positiva $s \leq 1$ és triat de tal manera que:

1. $f_i(z + s\Delta z) < 0$ per a tot $i = 1, \dots, m$,
2. El funcional ha minvat suficient:

$$f_0(z + s\Delta z) < f_0(z) + \alpha s \Delta z \langle g_z, \Delta z \rangle,$$

on α és un paràmetre. Un bon valor és $\alpha = 0.01$. Aquest punt ens diu que el decreixement ha d'estar a un cert percentatge el que ha predit el model lineal a z .

Com abans, començam amb $s = 1$ i decreixem amb múltiples de β mentre les prèvies condicions es satisfacin. Un bon valor per β és $\beta = 1/2$.

La implementació completa de l'algorisme log-barrera segueix el següent esquema:

1. Entrades: Un punt possible z^0 , una tolerància ν , i uns paràmetres μ i un τ^1 inicial. Definir $k = 1$.
2. Resoldre (7.7) mitjançant el mètode de Newton (seguit d'una cerca lineal per *backtracking*), emprant z^{k-1} com a punt inicial. Donar el nom z^k a la solució.
3. Si $m/\tau^k < \nu$, acabar i retornar z^k .
4. Si no, definim $\tau^{k+1} = \mu\tau^k$, $k = k + 1$ i tornar al punt 2.

De fet, podem calcular d'avançada quantes iteracions de l'algorisme log-barrera necessitarem:

$$\text{iteracions barrera} = \left\lceil \frac{\log m - \log \nu - \log \tau^1}{\log \mu} \right\rceil$$

El darrer problema és la tria de τ^1 . En el nostre cas, ho triarem de manera conservativa, és a dir, que després de la primera iteració el forat dual m/τ^1 és igual a $\langle c_0, Z^0 \rangle$.

7.2.2 Concreció de l'algorisme

El problema (7.5) es pot reescriure en la forma de SOCP,

$$\begin{aligned} \min_{t,x} \sum_{ij} t_{ij} \quad \text{tal que } \|D_{ij}x\|_2 \leq t_{ij}, \\ Ax = b. \end{aligned}$$

Si definim des funcions de desigualtat

$$f_{t_{ij}} = \frac{1}{2}(\|D_{ij}x\|_2^2 - t_{ij}^2) \quad i, j = 1, \dots, n,$$

tenim

$$\nabla f_{t_{ij}} = \begin{pmatrix} D_{ij}^T D_{ij} x \\ -t_{ij} \delta_{ij} \end{pmatrix}, \quad \nabla^2 f_{t_{ij}} = \begin{pmatrix} D_{ij}^* D_{ij} & \mathbf{0} \\ \mathbf{0} & -\delta_{ij} \delta_{ij}^T \end{pmatrix},$$

$$\nabla f_{t_{ij}} \nabla f_{t_{ij}}^T = \begin{pmatrix} D_{ij}^T D_{ij} x x^T D_{ij}^T D_{ij} & -t_{ij} D_{ij}^T D_{ij} x \delta_{ij}^T \\ -t_{ij} \delta_{ij} D_{ij}^T D_{ij} & t_{ij}^2 \delta_{ij} \delta_{ij}^T \end{pmatrix}.$$

on δ_{ij} és el vector de Kronecker que és 1 a l'entrada ij i zero a tots els altres llocs.

Si denotam per $\Sigma = \text{diag}(\{\sigma_{ij}\})$, D_h que té per columnes $D_{h;ij}$ (igualment per D_v), i B una matriu que depèn de x :

$$B = D_h^T \Sigma_{\partial h} + D_v^T \Sigma_{\partial v},$$

amb $\Sigma_{\partial h} = \text{diag}(D_h x)$, $\Sigma_{\partial v} = \text{diag}(D_v x)$.

El sistema de Newton (7.8) per l'algorisme de log-barrera queda llavors

$$\begin{pmatrix} H_{11} & B \Sigma_{12} & A^T \\ \Sigma_{12} B^T & \Sigma_{22} & \mathbf{0} \\ A & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \\ \Delta v \end{pmatrix} = \begin{pmatrix} D_h^T F_t^{-1} D_h x + D_v^T F_t^{-1} D_v x \\ -\tau Id - F_t^{-1} t \\ \mathbf{0} \end{pmatrix},$$

on

$$H_{11} = D_h^T (-F_t^{-1}) D_h + D_v^T (-F_t^{-1}) D_v + B F_t^{-2} B^T,$$

i definim

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} := \begin{pmatrix} D_h^T F_t^{-1} D_h x + D_v^T F_t^{-1} D_v x \\ -\tau Id - F_t^{-1} t \\ \mathbf{0} \end{pmatrix}.$$

Eliminant Δt ,

$$\begin{aligned} \Delta t &= \Sigma_{22}^{-1} (w_2 - \Sigma_{12} B^T \Delta x) \\ &= \Sigma_{22}^{-1} (w_2 - \Sigma_{12} \Sigma_{\partial h} D_h \Delta x - \Sigma_{12} \Sigma_{\partial v} D_v \Delta x), \end{aligned}$$

el sistema queda reduït a $(N + K) \times (N + K)$:

$$\begin{pmatrix} H'_{11} & A^T \\ A & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix} = \begin{pmatrix} w'_1 \\ \mathbf{0} \end{pmatrix}. \quad (7.9)$$

amb

$$\begin{aligned} H'_{11} &= H_{11} - B \Sigma_{12}^2 \Sigma_{22}^{-1} B^T \\ &= D_h^T (\Sigma_b \Sigma_{\partial h}^2 - F_t^{-1}) D_h + D_v^T (\Sigma_b \Sigma_{\partial v}^2 - F_t^{-1}) D_v \\ &\quad + D_h^T (\Sigma_b \Sigma_{\partial h} \Sigma_{\partial v}) D_v + D_v^T (\Sigma_b \Sigma_{\partial h} \Sigma_{\partial v}) D_h, \\ w'_1 &= w_1 - B \Sigma_{12}^2 \Sigma_{22}^{-1} w_2 \\ &= w_1 - (D_h^T \Sigma_b \Sigma_{\partial h} + D_v^T \Sigma_{\partial v}) \Sigma_{12} \Sigma_{22}^{-1} w_2, \\ \Sigma_b &= F_t^{-2} - \Sigma_{22}^{-1} \Sigma_{12}^2. \end{aligned}$$

El sistema d'equacions (7.9) és simètric, però no definit positiu. Not'is que D_h i D_v són matrius molt disperses i, llavors, poden ser emmagatzemades de manera molt eficient. Aquest fet ens capacita de nou a resoldre el sistema el problema esmentat utilitzant un mètode iteratiu com el SYMMLQ.

7.2.3 Rutines

A continuació podeu veure una breu explicació de què fa cada algorisme dels utilitzats el l'exemple que ens ocupa. El codi en Matlab de cada un d'ells el trobareu a l'annex B, on també hi haurà més informació de cada una de les rutines, així com dels paràmetres que se li han d'introduir i quins retorna.

recCLR.m

Donat un interval de mostres $[m, M]$ que es separarà segons la variable *skip* de tal manera que tendrem la partició $P = \{m, m + skip, m + 2skip, \dots, M - skip, M\}$, recupera la imatge en color "Lenna.jpg" per a cada valor de P . Amb la màscara lineal si *PoL* val 1 i amb la màscara uniforme si val 0. Els nombres n, M i *skip* corresponen al nombre de línies en el primer cas i al nombre de punts en el segon. Si la variable *fix* val 1, es fixarà la màscara uniforme per a totes les capes de color de cada recuperació. Retorna v , que és un vector amb el nombre de punts k utilitzats per a cada recuperació; Fs , l'error a cada capa de cada recuperació; *rec*, la recuperació de cada capa; i *ERROR*, la matriu diferència entre l'original i cada capa.

recMinTV.m

Donat un enter positiu n , un interval de mostres i un valor *skip*, crea el fantasma de Shepp-Logan (com a imatge de mostra), de dimensió $n \times n$, i llença la rutina de recuperació de la imatge agafant un nombre de mostres en l'interval donat, amb un pas de *skip* i utilitzant una màscara uniforme, i donant com a resultat la gràfica de l'error obtingut en la recuperació en funció del nombre de mostres. Alhora, també ens guarda la imatge recuperada i la matriu de l'error associada per a cada valor de de mostres pres.

recMinTV_lines.m

Donat un enter positiu n , un interval de mostres i un valor *skip*, crea el fantasma de Shepp-Logan (com a imatge de mostra), de dimensió $n \times n$, i llença la rutina de recuperació de la imatge agafant un nombre de mostres en l'interval donat, amb un pas de *skip* i utilitzant una màscara lineal radial, i donant com a resultat la gràfica de l'error obtingut en la recuperació en funció del nombre de mostres.

Ahora, també ens guarda la imatge recuperada i la matriu de l'error associada per a cada valor de de mostres pres.

tveq_phantom_example.m

Rep els paràmetres que ens indicaran si volem agafar un fantasma aleatori o del de Shepp-Logan, en el primer cas amb quantes el·lipses i en ambdós casos de quina dimensió; i la màscara que volem agafar, ja sigui uniforme, lineal a partir de radi, o una introduïda per l'usuari. El programa intenta recuperar la senyal bidimensional preparant els paràmetres adequats per a llençar la rutina *tveq_logbarrier.m*, prenent un nombre de mostres que l'usuari haurà d'introduir, en el cas que hagi triat una màscara uniforme o lineal, o que vendrà implícit si ha introduït una màscara pròpia. Retorna la imatge original, la imatge recuperada, la imatge d'energia mínima, la màscara utilitzada i l'error obtingut.

UniformRandomMask.m

Donat un enter N i un altre enter P , pren P punts de manera aleatòria sobre una imatge de $N \times N$, creant la màscara uniforme per a després prendre les mostres el altres programes.

LineMask.m

Donat un enter L i un altre enter P , pren L rectes passant per l'origen d'una imatge de $N \times N$, amb un angle igual entre elles, creant la màscara lineal radial, per a després prendre les mostres el altres programes.

A_fhp.m

Pren les mostres segons els coeficients que venen donats per Ω de la senyal x , i ho fa en la meitat del pla superior de la transformada de Fourier bidimensional. En el nostre cas, l'utilitzarem per a construir una matriu implícita.

At_fhp.m

Crea la matriu adjunta a l'obtinguda a *A_fhp.m*.

tveq_logbarrier.m

Rutina que resol el sistema de programació lineal $\min TV(x)$ tal que $Ax = b$ mitjançant un algorisme log-barrera passant-ho prèviament a un SOCP.

tveq_newton.m

Subrutina que fa una iteració de Newton per a l'algorisme log-barrera que realitza *tveq_logbarrier.m*.

randomPhantom.m

Donat un enter dim i un nombre d'el·lipses n , crea una imatge de $dim \times dim$ i hi pinta n el·lipses prenent els seus radis i eixos de forma aleatòria, així com també el seu color, que serà alguna tonalitat de gris.

ellipseMatrix.m

Donades les dades que descriuen una el·lipse, la pinta en la imatge donada. En el cas de que ja hi hagi altres el·lipses pintats a la imatge donada, pren les tonalitats de gris de la intersecció com a la suma de tonalitats, que estan entre 0 i 1, mòdul 1.

is_val.m

Donats uns certs paràmetres que descriuen una el·lipse i un punt, retorna si el punt està dintre de la el·lipse descrita.

7.2.4 Resultats

Separarem aquest exemple en dues parts, tot treballant en imatges, senyals bi-dimensionals. Pel que fa a la primera, prendrem com a exemple el fantasma de Shepp-Logan. No esperam trobar una fita per al nombre de mostres, ja que els resultats que hem desenvolupats, tot i ser extensibles a dimensions superiors, canviarien lleugerament. L'objectiu d'aquest exemple és demostrar que la recuperació de senyals de dimensió 2 és possible. De la mateixa manera, provar diferents màscares de mostreig i treure conclusions sobre la recuperació en cada una d'elles. En la segona part, reconstruirem una imatge en color, comparant les distintes màscares i traient conclusions segons l'elecció.

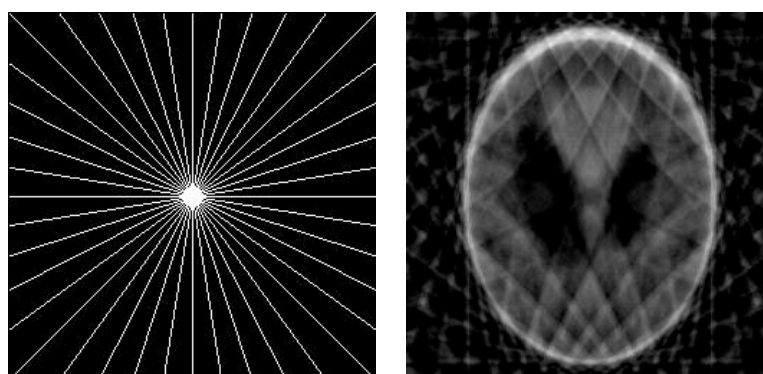


Figura 7.6: Imatge de mostra. Fantasma de Shepp-Logan.

La imatge que prendrem de mostra serà el fantasma de Shepp-Logan que podem veure a la figura 7.6. El prenem de 256×256 , és a dir, $N = 65536$. Com sabem, amb el mostreig de compressió podem recuperar imatges a partir de poques mostres. A medicina, per exemple, a l'hora de realitzar una tomografia, és troben amb molt poques mostres a espai de Fourier de la imatge que es vol obtenir. Precisament, es pren aquest gràfic perquè és semblant a una tomografia, que no és més que una imatge que representa un tall transversal del cos humà. De manera opcional, l'usuari pot introduir un fantasma creat per la superposició de E el·lipses amb paràmetres aleatoris utilitzant la funció que hem creat per a tal efecte, `randomPhantom.m` o qualsevol altra imatge.

La rutina per a obtenir la recuperació s'anomena `tveq_phantom_example.m`, on per tal de prendre les mostres utilitzarem dos tipus de màscara², segons les subrutines emprada, `LineMask.m` o `UniformRandomMask.m`. En el primer, es crea una màscara

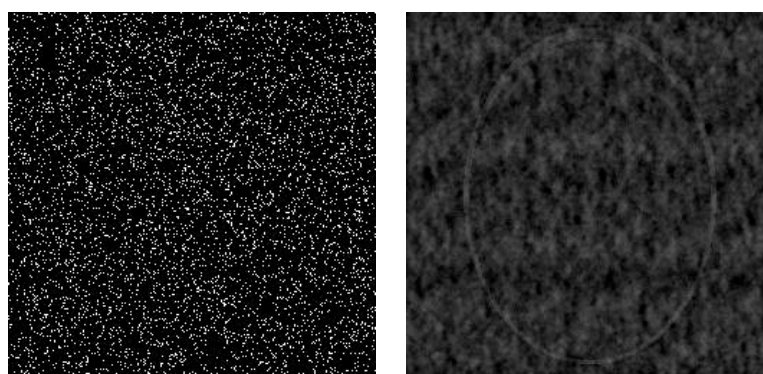
²A l'adreça web <https://mega.co.nz/#!M81ykBYB!YHalbiRBhv159iyBYWqFJhiAVI8WJiyXVsA6Cfcf-I> podreu trobar un vídeo en el que apareixen les dues màscares de mostreig.



(a) Màscara, 5001 punts (equivalent al 7.6% del total). (b) Recuperació de mínima energia. Mostreig realitzat.

Figura 7.7: Màscara i mínima energia obtinguda després de prendre les mostres.

formada per un nombre de línies L , introduït com a paràmetre, col·locades de manera radial passant pel centre de la imatge. El nombre de punts presos per aquesta màscara es calcula posteriorment amb l'ajuda de la funció $K=\text{nnz}(X)$ del Matlab, que retorna el nombre de posicions distintes a zero de la matriu X . En el nostre cas, la matriu que representa la màscara lineal. En el segon, s'agafen K punts presos de manera aleatòria seguint una distribució uniforme.



(a) Màscara, 5400 punts (8.2%). (b) Recuperació de mínima energia.

Figura 7.8: Màscara i mínima energia obtinguda després de prendre les mostres.

Com que aquest exemple és a gran escala, ja que el nombre de punts involucrats és molt gran, utilitzam matrius implícites per a manejar les mostres. Notem que el sistema resultant és de $K \times 65536$. Ho feim mitjançant les funcions `A_fhp.m` i `At_fhp` i l'eina per a funcions implícites que té el Matlab. El primer mètode dels esmentats pren la imatge de 256×256 i retorna un vector de mostres en

les K freqüències. Com que la imatge és real, la rutina retorna les parts reals i imaginàries de la transformada ràpida de Fourier en dues dimensions en la meitat superior del pla donat per la màscara triada. El segon retorna la transposada de l'anterior. Així, el programa entén les matrius com a funcionals, que aplicat a la imatge reescalada com a vector ens donarà com a resultat el mostreig.

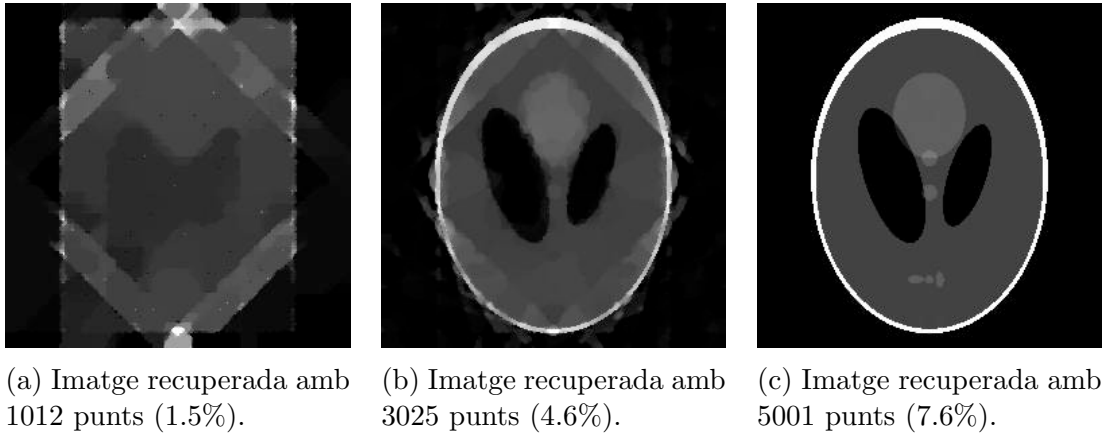


Figura 7.9: Resultat de la recuperació amb la màscara lineal .

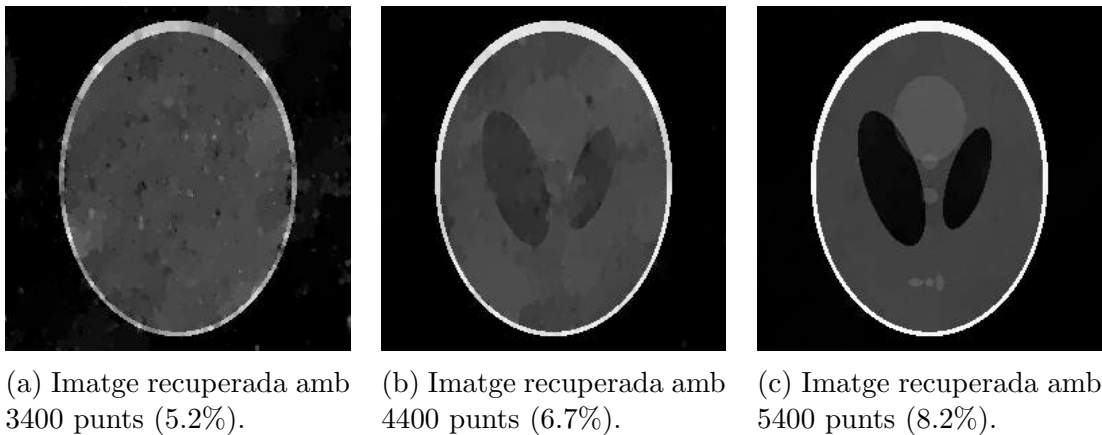


Figura 7.10: Resultat de la recuperació amb la màscara uniforme.

A la figura 7.7a es pot observar la màscara lineal corresponent a prendre 20 línies, que resulten ser $K = 5001$ punts, un 7.6% del total de punts. A 7.7b hi ha la recuperació que genera el programa a partir dels punts obtinguts en el mostreig amb la màscara donada. Anàlogament, a la figura 7.8a i 7.8b es mostren la màscara uniforme consistent en $K = 5400$ punts, equivalent a un 8.2% del total de punts, i la recuperació de mínima energia respectivament.

La recuperació de mínima energia s'utilitza en la rutina `tveq_logbarrier.m` com a punt inicial en l'algorisme de punt interior log-barrera, de la mateixa manera que s'hi introdueix el mostreig $y = A(x)$ i algunes dades referents a l'error amb que volem els resultats de l'algorisme, del nombre de passos màxims de Newton i de l'error de Newton per a la subrutina `tveq_newton.m`.

Com veim en les figures 7.9³ i 7.10⁴, en ambdós casos obtenim la recuperació per als valors de K prèviament esmentats. En aquestes figures podem veure com millora el resultat obtingut així com anem afegint punts quan prenem el mostreig.

Pel que fa a les màscara lineal, a la imatge 7.9a hem pres 4 línies, corresponents a 1012 punts, un 1.5% del total; en 7.9b 12 línies, corresponents a 3025 punts, un 4.6% del total; i en 7.9c 20 línies, corresponents a 5001 punts, un 7.6% del total. De la mateixa manera, en les imatge 7.10a, 7.10b i 7.10c, trobem les recuperacions per a valors de K corresponents a 3400, 4400 i 5400 punts, un 2.2%, 6.7% i 8.2% del total respectivament.

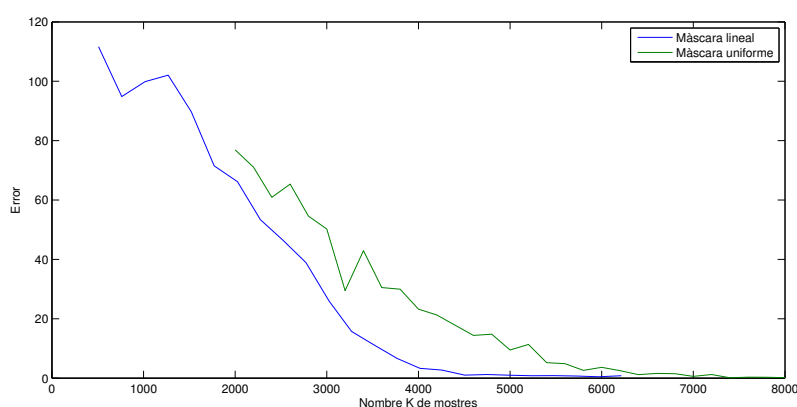


Figura 7.11: Error en la recuperació en funció del nombre de punts i de la màscara.

La recuperació s'aproxima tant a l'original que l'error no és apreciable a simple vista. Per tal de veure la comparativa entre la imatge recuperada i l'error respecte a l'original es remet al lector a l'annex C. De la mateixa manera, prenent la norma matricial $\|A\| = \max_j \sum_i |a_{ij}|$, obtenim el gràfic de la figura 7.11, on hi ha representat l'error, $\|X - X_{tv}\|$, entre la imatge original X i la recuperada X_{tv}

³A l'adreça web https://mega.co.nz/#!1g8xhJyK!b8P3Uy-lrAbLu3GkBiMgPi7Mh_MvaLJO75I20GJo3Vs podeu trobar un vídeo on es pot veure la recuperació lineal.

⁴A l'adreça web <https://mega.co.nz/#!Zh0G1CTa!Ga97GT-0FqxlsazEAXYpOwvYkg4EHG-cuZTdh6lxEC90> podeu trobar un vídeo on es pot veure la recuperació uniforme.

en funció del nombre de mostres preses i de la màscara. Hem obtingut aquestes dades mitjançant les rutines `recMinTV.m` i `recMinTV_lines.m`.

En aquest cas, notem que la recuperació és possible amb menys mostres si prenem la màscara lineal. Això és resultat de la forma que té la màscara. En el cas de la lineal hi ha molts de punts en la part central i per tant hi haurà més informació acumulada. En el cas de la màscara uniforme la informació és més dispersa. Tot i així, la diferència no és massa gran i la imatge és recuperada perfectament.

Com a observació final, i potser la més important, és que les màscares no tenen cap relació amb la imatge, en són totalment independents. Per tant, els resultats que hem obtingut amb aquesta imatge poden ser directament extrapolats a qualsevol imatge de la mateixa dimensió. El temps de còmput per a una imatge de 256×256 píxels amb l'algorisme utilitzat és d'aproximadament 700 segons. Si prenem la imatge de 512×512 aquest temps s'incrementa fins als 6500 segons.

En color

Què passa amb les imatges en color? Com és ben sabut, una imatge en color és la superposició de tres capes –vermella, verda i blava– que es poden pensar com si estassin en tons de grisos, ja que cada una es representa per una matriu bidimensional amb valors dins la mateixa escala que les imatges en blanc i negre. Aquestes tres capes es representen com una matriu tridimensional, superposant cada píxel. D'aquesta manera, l'ordinador representa el color –com a llum, no com a pigment– com a una mescla de vermell, verd i blau tot mirant el valor de cada una de les capes a cada punt.

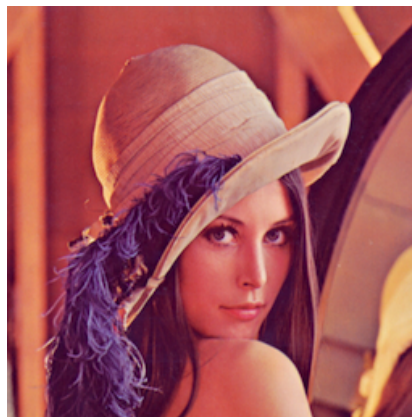


Figura 7.12: Imatge de mostra en color. Fotografia de na Lenna.

Per tant, és lògic pensar que podem aplicar el procediment anterior a imatges en color aplicant-lo a cada capa en particular. Per provar l'eficiència de les rutines anteriors hem triat una imatge que correspon a una fotografia de na Lenna. La imatge de 256×256 la podeu trobar a la figura 7.12. Hem triat aquesta imatge per la seva riquesa en textures i degradats, molt distinta al fantasma de Shepp-Logan, que tenia gradients molt accentuats entre diferents blocs de colors sòlids.

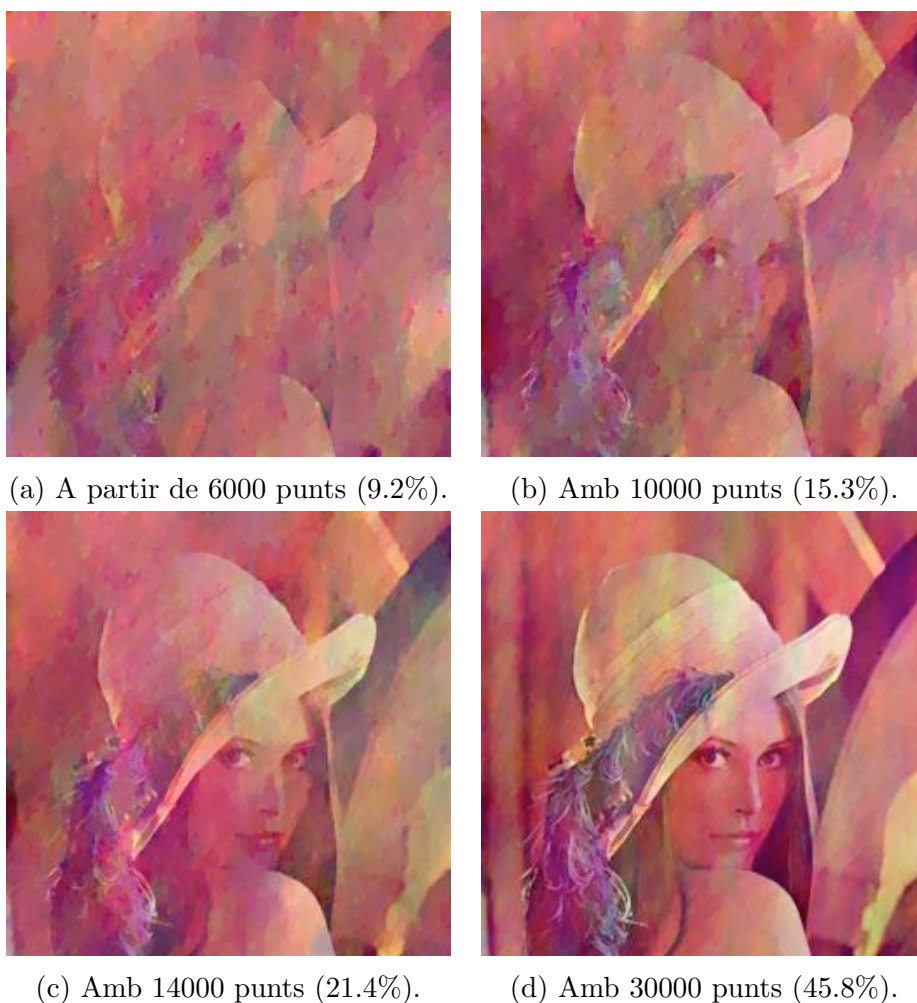


Figura 7.13: Resultats de la recuperació amb la màscara uniforme.

Tornarem a emprar el mateix algorisme de programació lineal amb la funció $TV(x)$. La primera prova correspondrà a emprar la màscara uniforme. Emprarem la rutina `recCLR.m`, la qual per a cada capa de color i cada valor de mostres K crida a la rutina `recMinTV.m`. Així com programarem aquesta darrera, crea una màscara uniforme de manera aleatòria, així que el mostreig canviarà per a cada valor de K però també per a cada capa de color.

A la figura 7.13 podem veure la recuperació per a diferents valors de K . Hi ha dos problemes: el primer el colors, que semblen mesclats, i el segon, que són necessàries moltes mostres i tanmateix, per moltes que n'agafem, no s'obtenim bones recuperacions.

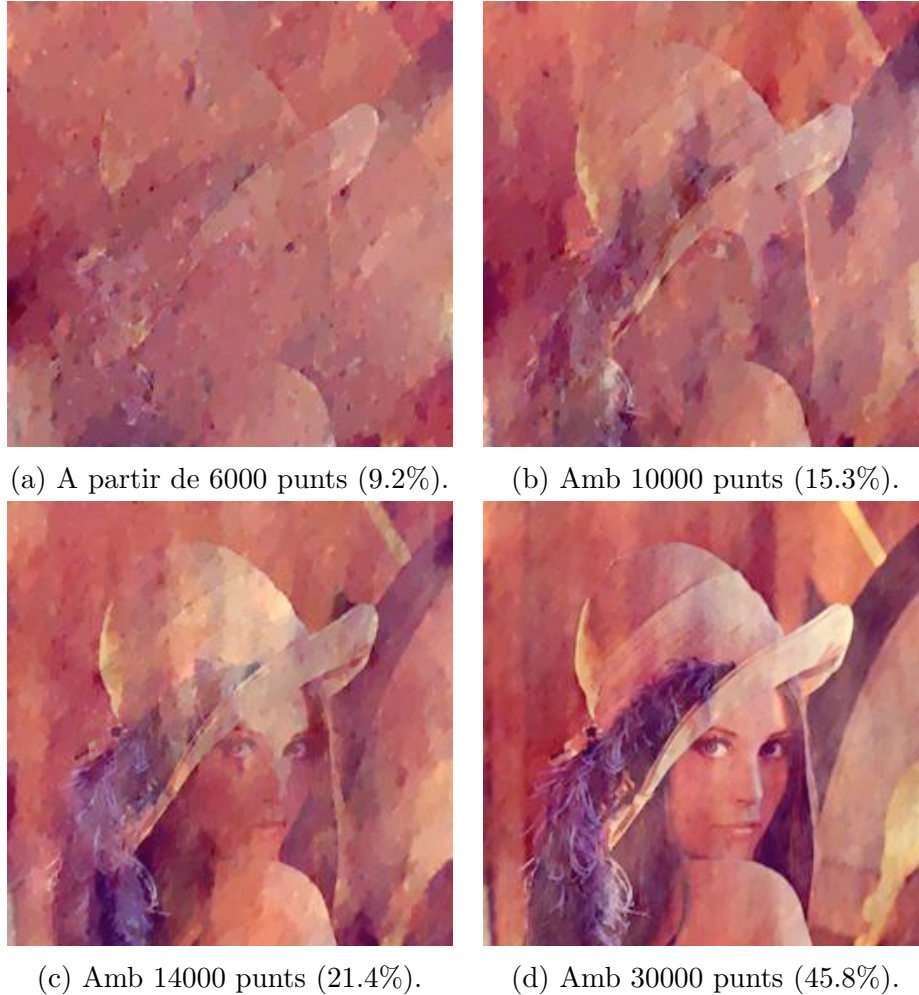


Figura 7.14: Resultats de la recuperació amb la màscara uniforme fixada.

La primera hipòtesi sobre el problema del color és que prenem les mostres per a cada color de manera independent. Per tant, com que la màscara de mostreig no és iguala, les aproximacions que s'obtenen per a cada capa no coincideixen del tot, i en resulten colors que no es troben en les tonalitats de la imatge original. Això ho podem resoldre fixant la màscara per a cara recuperació. Així, creem una màscara amb K punts presos amb una distribució uniforme i l'emprarem per dur a terme el mostreig per a totes les capes de colors. El resultat obtingut és el que podem veure a la figura 7.14.

En aquestes imatges, podem veure com la imatge va millorant així com anam afegint punts a la màscara de mostreig, i que els colors estan dins la mateixa paleta que els originals. Tot i així, la recuperació és molt dolenta inclús si agafem el 45.8% dels punts.



Figura 7.15: Resultats de la recuperació amb la màscara lineal.

Si ens fixem amb la imatge 7.14d, es veuen unes bandes verticals separades per un canvi de color bastant notori. Comparant-ho amb la imatge original, la de la figura 7.12, aquestes bandes coincideixen amb el canvis d'il·luminació en els quals existeix un degradat. Recordem que la màscara uniforme pren informació de punts repartits per tota la imatge per condensar-ho en un punt de l'espai de freqüències. Per tant, la funció $TV(x)$, que és una espècie de gradient, no detecta bé els canvis de tonalitat.

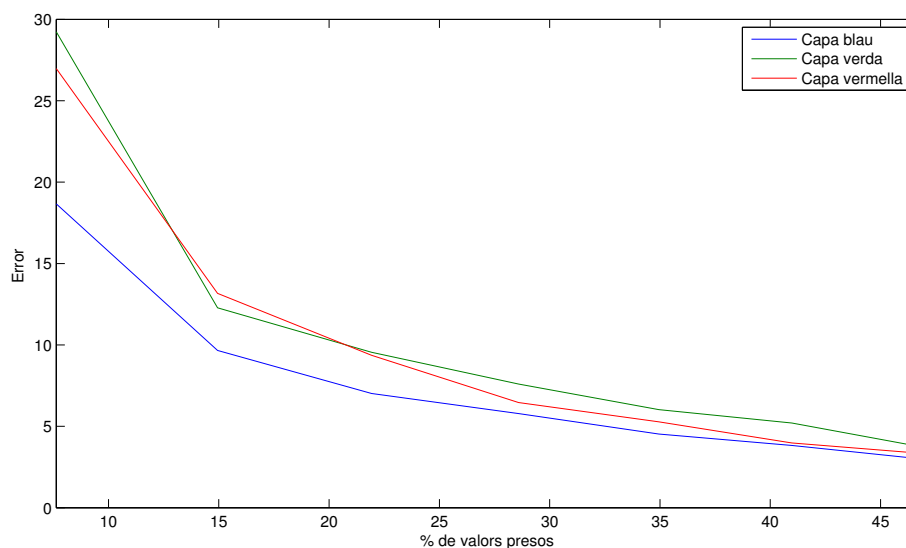


Figura 7.16: Error en funció del percentatge de mostres i de la capa per a la màscara lineal.

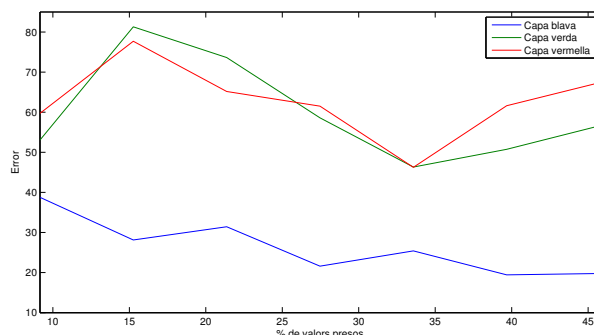
Per altra banda, la màscara lineal, per condensar la informació en un punt de l'espai de freqüències, la pren de manera radial en un entorn del punt, i per tant hi és més patent els canvis que la imatge sofreix en aquell entorn. D'aquesta manera, com podem observar en la figura 7.15, la recuperació que aconseguim és molt acurada amb aproximadament un 20% dels punts, i quasi exacta amb el 35%.

Observem un fet singular. La millora entre la imatge de la figura 7.15a i la de la figura 7.15b, amb una diferència d'un 7% de valors, és molt més gran que la diferència entre les figures 7.15c i 7.15d, amb una diferència del 13% de valors. Per tant, la millora més gran es dona amb pocs valors, i després en són necessaris bastants per a anar refinant la imatge. Aquest efecte prové del fet que amb la base de Fourier la informació s'emmagatzema en pocs punts i que, un cop agafada, la resta d'informació –menys rellevant– està molt repartida en molts punts.

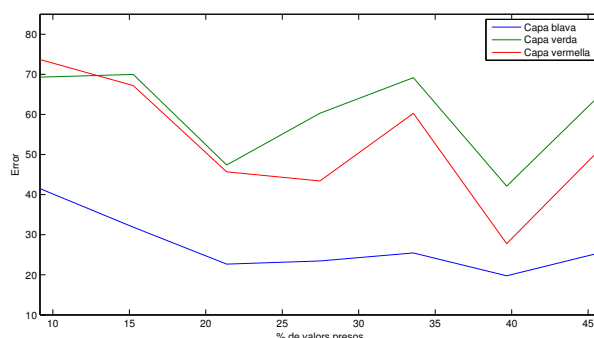
El fet prèviament esmentat es pot veure a la figura 7.16, on podem observar com disminueix l'error de cada capa depenent del percentatge de punts que prenem de mostra per recuperar la imatge. Notem que no s'ha d'entendre l'error de la imatge en color com a la suma de l'error de les diferents capes, sinó com a una mitja.

Per altra banda, com hem observat en les figures 7.13 i 7.14, la recuperació no és bona per a cap valor de K bastant menor de N , ja sigui per les tonalitats de color

com pels canvis de gradient que provoquen. Els gràfics corresponents a l'error els trobem a les figures 7.17a i 7.17b.



(a) Amb la màscara aleatòria per capes.



(b) Fixant la màscara per capes.

Figura 7.17: Error en funció del percentatge de mostres i de la capa per a la màscara uniforme.

Per tant, podem concloure que si volem minimitzar la funció $TV(x)$ necessitam prendre la informació en l'espai de Fourier condensant la informació en l'entorn del punt, per tal de poder detectar els canvis de gradient.

De totes maneres, comparant els resultats obtinguts per imatges de amb blocs de color com és el fantasma de Shepp-Logan i imatges amb degradats com és la fotografia de na Lenna, aquest programa és molt millor per les primeres, ja que aconseguix una reconstrucció més acurada. Per les primeres era amb un 7.6% de mostres i un error menor a 1. En canvi, per la fotografia aconseguíem una reconstrucció acceptable prenent el 21.9% dels coeficients i amb un error notable, d'unes 7 unitats. Segons la imatge a reconstruir, convendria utilitzar aquest mètode o desenvolupar altres funcions que detectin millor els degradats.

Capítol 8

Conclusions

La teoria que hi ha sota el mostreig de compressió, tot i ser de total actualitat, es fonamenta en resultats sobre teoria de la informació que es remunten als anys seixanta. La suma de les parts d'una manera senzilla, tot i que no trivial, i harmònica, així com la seva utilitat en qualsevol camp on hi intervingui l'adquisició o transmissió de senyals, aporta a l'estudi una gran importància. Els motius que m'han duit a triar aquest tema per a desenvolupar el treball han estat aquests, així com el meu interès en treballar en l'àrea de l'anàlisi harmònic relacionada amb el tractament d'imatges.

Per sobre del Teorema fonamental de l'àlgebra i del Teorema de mostreig de Nyquist-Shannon, amb la teoria desenvolupada s'aconsegueix, sota unes certes condicions, reconstruir una senyal a partir d'un percentatge molt petit de mostres.

Una de les preguntes que ens formulàvem abans de començar era quin nombre mínim de mostres hem de prendre per tal de reconstruir la senyal prou bé. Aquesta pregunta passa per definir primer com hem de prendre les mostres per tal de minimitzar aquest nombre.

Recordem que empràvem una certa base Φ , de detecció, per a prendre les mostres i una altra Ψ , de representació, per a treballar amb el conjunt de mostres, tot accentuant la dispersió de la imatge. D'aquesta manera, definíem el coeficient $\mu(\Phi, \Psi)$ de correlació que, recordem, intervé directament en el nombre de mostres pres i que, com més petit sigui, millor. D'altra banda, les senyals d'interès normalment són S -disperses en l'espai on les prenem, és a dir, que tenen S components distintes a zero. Aquest valor és un altre factor que intervé, conjuntament amb el

nombre total d'elements de la senyal en la fórmula donada pel Teorema 5.1.5,

$$m \geq C_M \cdot \mu^2(\Psi, \Phi) \cdot S \cdot \log n, \quad (8.1)$$

i que ens assegura que si hem pres m mesuraments de forma uniformement aleatòria tal que compleix (8.1) podrem recuperar la senyal de dimensió n amb una probabilitat de $1 - O(n^{-M})$. El factor C_M depèn de la probabilitat que vulguem. Notem, que per petit que sigui M , la probabilitat es dispara ràpidament cap a 1. Com ha quedat demostrat en la secció 5.2, hi ha un límit fonamental, és a dir, que el nombre de mostres m està fitat inferiorment, i que aquesta fita depèn de les bases que estem utilitzant. En la figura 7.5 es pot observar la comparació del resultat teòric amb l'empíric.

Reconstrucció en probabilitat *versus* determinista

El resultat que primer hem desenvolupat són probabilístics. Es basen en l'existència d'un polinomi dual,

$$P := \mathcal{F}_\Omega^* \mathcal{F}_{T \rightarrow \Omega} (\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega})^{-1} t^* \operatorname{sgn} f,$$

amb unes certes propietats,

1. *Invertibilitat*. L'operador $\mathcal{F}_{T \rightarrow \Omega}$ és injectiu. Per tant, el terme $\mathcal{F}_{T \rightarrow \Omega}^* \mathcal{F}_{T \rightarrow \Omega}$ de (4.1) és invertible.
2. *Magnitud en T^c* . La funció P en (4.1) satisfà $|P(t)| < 1$ per a tot $t \in T^c$.

les quals compleix amb una probabilitat de $1 - O(n^{-M})$. Demostrat això, el Teorema 5.1.5, amb ajuda del Lema 5.1.1, ens asseguren el resultat amb la probabilitat esmentada.

Gràcies als resultats probabilístics podem fitar el nombre de mostres depenent de certs factors com les bases en les que treballem, el tipus de senyal a tractar, la mida d'aquesta i la probabilitat d'error que ens puguem permetre. Però els resultats que s'han demostrat aquí van molt més lluny, s'han estudiat sota quines condicions es pot obtenir la recuperació amb resultats deterministes, sense que hi intervinguin probabilitats.

A partir de les constants d'isometria restringida i d'ortogonalitat, resultats com el Lema 3.3.5 ens asseguren la reconstrucció de la senyal, però d'una forma teòrica. En el cas probabilístic sabíem que la podíem reconstruir i com. Però hem anat

més enllà i , utilitzant la propietat de reconstrucció exacta, s'ha demostrat en el Teorema 6.3.1 que la reconstrucció és possible, i com trobar-la, quan existeix la següent relació entre les constants d'isometria i d'ortogonalitat restringida,

$$\delta_S + \theta_S + \theta_{S,2S} < 1.$$

A més, hem vist les matrius aleatòries de Gauss tenen aquestes bones propietats i que, per tant, són susceptibles de ser emprades per a la reconstrucció d'imatges.

En ambdós casos, el mostreig de senyals es fa de manera totalment independent a la senyal i , per tant, es pressuposa un coneixement nul sobre aquesta i sobre els valors on es condensa la majoria d'informació.

Empíricament

En la secció 7 hem comprovat com la teoria realment es pot dur a la pràctica i que els resultats són certs dins els marges que ens donen les hipòtesis establertes.

Per a les senyals unidimensionals, hem vist en el gràfic de la figura 7.5, com el Teorema 5.1.5 es compleix. En aquest gràfic es pot comprovar el resultat per a distints valors de S , que recordem que és el nombre de valors distints a zero de la senyal.

Per a les senyals bidimensionals amb blocs de grisos i gradients molt marcats aconseguim la recuperació prenent sols un 7.6% de mostres, amb un error menor a la unitat. En canvi, per a imatges amb gradients suaus, aconseguim una reconstrucció acceptable prenent el 21.9% dels coeficients i amb un error notable, d'unes 7 unitats. Les imatges en color es recuperen recuperant cada capa de color –vermella, verda i blava– separatament tractant-les com si fossin en blanc i negre.

Millores en resultats teòrics

Tal com afirmen Candès i Tao en [Can08], són molts els resultats teòrics que s'han vist afectats per la teoria que ells, juntament amb altra gent, desenvoluparen.

Per exemple, respecte la geometria d'alta dimensió que hi ha implícita en el mostreig de compressió, hi ha unes quantes puntes a tenir en compte. Primer, que el mostreig de compressió ve amb un algorisme –minimització ℓ_1 – de tal manera que els resultats de la teoria sobre espais de Banach no és sols existencial sinó pràctica, ja que hom pot recuperar els objectes d'estudi mitjançant la programació lineal.

Cal remarcar aquí el fet que el mateix algorisme resulta ser quasi òptim sobre moltes classes de senyals d'interès, concretament les molt emprades bolles en ℓ_p per a $p < 1$. Per altra banda, ens atorga una altra manera de veure la geometria dels espais de Banach. El Teorema de Kashin [Kas77], el qual es considerava extremadament profund i amb un argument difícilment penetrable passà de tenir una prova d'entre 100 i 200 pàgines a no més de 2 pàgines que un estudiant amb coneixements bàsics d'àlgebra lineal i de probabilitats podria entendre. Finalment, les noves tècniques i eines permeten abordar altres preguntes sobre algorísmia i teoria de la informació com la robustesa de la reconstrucció juntament amb el sorolls i la quantització d'errors.

Millors en els algorismes

El mostreig de compressió ha d'anar de la mà dels algorismes de programació lineal per a resoldre de manera eficient problemes ℓ_1 a gran escala. De fet, a partir del desenvolupament d'aquesta teoria s'han anat optimitzant les rutines existents i cercant-ne altres d'alternatives des d'altres punts de vista, prenent idees de la matemàtica discreta o de la teoria de grafs.

Aplicacions

Actualment, el treball de Candès i Tao està motivat pel problema de reconstruir imatges biomèdiques a partir d'un mostreig pobre en l'espai de Fourier. Problemes d'interès són el de la ressonància magnètica per angiografies, però pot ser estès a qualsevol altre tipus de ressonància magnètica, o d'adquisició d'imatges, com les tomografies.

També canviarà la manera de pensar sobre les òptiques digitals, ara tenim càmeres d'un sol pixel, i sembla que el mostreig de compressió ajudarà a implementar noves arquitectures de compressió d'imatges en CMOS.

Una altra aplicació seria entendre aquest procés de mostreig/reconstrucció com un tipus de codificador/descodificador. En aquest cas, el mostreig jugaria el paper de codificador i el programa lineal d'optimització de descodificador.

Apèndix A

Codi font: exemple 7.1

A.1 Valors de recuperació mínima en $\ell_1(x)$

```
0 function [vars ,F,G,MU]=recMin(nk, nK, nt,nT, N, M, enMitja)
  % Si nt=nT retorna el gràfic de l'error dependent K.
  % Si són diferents, retorna per a quin valor de K es
  % té "sempre" la recuperació exacta per a cada valor de T.

5 % INPUT:
  % [nk,nK] - interval de presa de mostres
  % [nt,nT) - nombre de puntes
  % N       - Mida de la imatge
  % M       - Paràmetre d'error
10 % enMitja - 0 (gràfic teòric amb cada MU independent).
  %          1 (gràfic teòric amb la mitja de les MU).
  %          S'obvia si nt=nT.

  % OUTPUT: (si nt=nT)
15 % vars - valors de K presos (si nt=nT) o de T (si nt/=nT).
  % F    - valor de K es té "sempre" la recuperació.
  % G    - [] (retorna un vector buid)
  % MU   - valor "real" de MU.

20 % OUTPUT: (si nt~nT)
  % vars - valors de T presos.
  % F (T) - m mínim de recuperació segura.
```

```

% G (T) - primers valors de m amb recuperació exacta.
% MU (T) - valors "real" de MU per a cada T.
25
% EXEMPLES:
% [vars, f, g, MU]=recMin(2, 150, 1, 35, 512, 1.3, 1);
% [vars, f, g, MU]=recMin(2, 150, 10, 10, 512, 1.3, 0);
30 %CONDICIO (1.5 - 0.87*(M+ 1))*log(N)<-3.10
if ((1.5 - 0.87*(M+ 1))*log(N)>=-3.10),
    disp('A necessary condition is not fulfilled!');
    disp(['n=' , num2str(N) , ' ; M=' , num2str(M)]);
end
35
%Inicialitzam les variables i certs valors
n=nK-nk;
n2=nT-nt; if n2==0, n2=1; end;
corba=zeros(1,n);
40 K_rec_exacta = zeros(1,n2);
K_primera_rec=zeros(1,n2);
FITES = zeros(1,n2);
C = 23*(M+1);
MU = ones(1,n2);
45 logN=log(N);

for j=1:n2;
    exit=0;
    for i=n:-1:1,
50 [xp, ERROR, mu] = lleq_example(nt+j-1,nk+i-1,N,M);
    corba(i)=sqrt(sum(ERROR.*ERROR));
    if (corba(i)<0.5),
        K_primera_rec(j)=nk+i-1;
    end
55 if (exit==0) && (corba(i)>0.5),
        MU(j)=mu;
        K_rec_exacta(j) = nk+i;
        FITES(j) = C*MU(j)*MU(j)*(nt+j-1)*logN;
        exit=1;
60 end
    end
end
end

```



```

65 if nt==nT,
    MU=1;
    varsK=nk:(nK-1);
    figure ,
    plot(varsK , corba );

70    disp(sprintf( 'M=_%2.1f ' ,M ));
    prob = 1-N^(-M);
    disp(sprintf( 'Prob=_%1.4f ' ,prob ));
    disp( ' ' );
    C = 23*(M+1);
75    disp(sprintf( 'C=_23*(M+1) _=%5.5f ' ,C ));
    disp(sprintf( 'MU=_%5.1f ' ,MU ));
    disp(sprintf( 'S=_%5i ' ,nT ));
    disp(sprintf( 'n=_%5i ' ,N ));
    FITA = C*MU^2*nT*log(N);
80    disp( ' ' );
    disp(sprintf( 'C_· _MU^2_· _S_· _log(n) _=%5.2f ' ,FITA ));
    vars=varsK ;
    F=corba ;
    G=[];
85 else
    varsT=nt:(nT-1);
    if enMitja==1,
        Mu=MU; Mu=sum(MU)/(length(MU));
        FITES=Mu.*Mu.*varsT*log(N)*C;
90    end

    figure ,
    plot(varsT , K_rec_exacta , varsT , K_primera_rec , ...
                                                varsT , FITES);
95    vars=varsT ;
    F=K_rec_exacta ;
    G=K_primera_rec ;
end

```

A.2 Recuperació mínima en $\ell_1(x)$

```

0  % l1eq_example.m
   %
   % Test out l1eq code:
   % (l1 minimization with equality constraints).

5  % Written by: Justin Romberg, Caltech
   % Email: jrom@acm.caltech.edu
   % Created: October 2005
   % Modified: Sebastià Pons Burguera, may 2013

10 function [xp, ERROR, MU] = l1eq_example(T,K,N,M)
   % INPUT:
   % (without inputs, then interactive routine)
   % (with ALL inputs, to use as a subroutine)
   % T – Nombre de mostres
15  % K – Nombre de pics (Coeficient de dispersió)
   % N – Mida de la senyal
   % M – Paràmetre d'error

   % OUTPUT:
20  % xp      – Senyal reconstruïda.
   % ERROR – Matriu diferència entre original i reconstruïda.
   % MU      – Valor "real" de MU.

   % EXEMPLES:
25  % l1eq_example;
   % [xp, ERROR, MU] = l1eq_example(20,120,512,1);

   if (nargin == 0),
       % signal length
30       N = 512;
       % number of spikes in the signal
       T = 20;
       T = input('T=_ '); disp(' ');
   end;

35  % random +/- 1 signal
   x = zeros(N,1);

```

```

q = randperm(N);
x(q(1:T)) = sign(randn(T,1));
40
% values to display later
if (nargin > 0),
    prob = 1-N^(-M);
    C = 23*(M+1);
45    FITA = C*T*log(N);
end

if (nargin == 0),
    % Displaying parameters
50    M = 1.0;
    disp(sprintf('M=%2.1f',M));
    prob = 1-N^(-M);
    disp(sprintf('Prob=%1.4f',prob));
    disp(' ');
55    C = 23*(M+1);
    disp(sprintf('C=%23*(M+1)=%5.5f',C));

    % Creating MU approximation taking K 20% of N.
60    A=random('unif',-ones(ceil(N/5),N),ones(ceil(N/5),N));
    A = orth(A)';
    for i=1:ceil(N/5), B(i)=sum(A(i,:))/sqrt(N); end;
    MU = max(B);
    %
    disp(sprintf('~MU=%5.1f',MU));
65    disp(sprintf('S=%5i',T));
    disp(sprintf('n=%5i',N));
    FITA = C*MU^2*T*log(N);
    disp(' ');
    % Displaying an approximate infimum for K
70    disp(sprintf('C * MU^2 * S * log(n) = %5.2f',FITA));
    disp(' ');

    % number of observations to make
75    K = 120;
    K = input('m= '); disp(' ');
end;

```

```

% Creating measurement matrix
80  if (nargin == 0),
        disp(' ');
        disp('Creating measurement matrix ... ');
    end
% We take uniform random matrix
A = random('unif',-ones(K,N),ones(K,N));
85  %If we want to take it as N(0,1) uncomment:
%A = randn(K,N);

% orthogonalization of the matrix
90  A = orth(A)';

% Computing real MU & the real value for the infimum
for i=1:K, B(i)=sum(A(i,:))/sqrt(N); end;
MU = max(abs(B));

95  %Display values
    if (nargin == 0),
        disp('Done. ');
        disp(' ');
        disp(sprintf('Real MU = %5.1f',MU));
100  FITA = C*MU^2*T*log(N);
        disp(sprintf('Fita real = %5.2f',FITA));
    end

% Observations
105  y = A*x;

% Initial value = min energy
x0 = A'*y;

110 % Solving LP
    if (nargin == 0),
        tic
    end
xp = l1eq_pd(x0, A, [], y, 1e-3);
115  if (nargin == 0),
        toc
    end
end

```

```

120 % large scale
% Afun = @(z) A*z;
% Atfun = @(z) A'*z;
% tic
% xp = l1eq-pd(x0, Afun, Atfun, y, 1e-3, 30, 1e-8, 200);
% toc
125
% Error computation
ERROR=abs(xp-x);

% Plot of the results
130 if (nargin == 0),
    figure,
    subplot(2,2,1); plot(x); axis([0 N -1 +1]);
    subplot(2,2,2); plot(x0); axis([0 N -1 +1]);
    subplot(2,2,3); plot(xp); axis([0 N -1 +1]);
135 subplot(2,2,4); plot(ERROR); axis([0 N 0 max(ERROR)*2]);
end

```

A.3 Solucionador de programació lineal (primal-dual)

```

0 % l1eq-pd.m
%
% Solve
% min_x ||x||_1 s.t. Ax = b
%
5 % Recast as linear program
% min_{x,u} sum(u) s.t. -u <= x <= u, Ax=b
% and use primal-dual interior point method
%
% Usage:
10 % xp = l1eq-pd(x0, A, At, b, pdtol, pdmaxiter, cgtol, ...
%                                     cgmaxiter)
%
% x0 - Nx1 vector, initial point.
%

```

```

15 % A – Either a handle to a function that takes a N vector
%     and returns a K vector , or a KxN matrix. If A is a
%     function handle, the algorithm operates in
%     "largescale" mode, solving the Newton systems
%     via the Conjugate Gradients algorithm.
20 %
% At – Handle to a function that takes a K vector and
%     returns an N vector. If A is a KxN matrix,
%     At is ignored.
%
25 % b – Kx1 vector of observations.
%
% pdtol – Tolerance for primal–dual algorithm (algorithm
%     terminates if the duality gap is less than pdtol).
%     Default = 1e–3.
30 %
% pdmaxiter – Maximum number of primal–dual iterations.
%     Default = 50.
%
% cgtol – Tolerance for Conjugate Gradients; ignored if A
35 %     is a matrix. Default = 1e–8.
%
% cgmaxiter – Maximum number of iterations for Conjugate
%     Gradients; ignored if A is a matrix.
%     Default = 200.
40 %
% Written by: Justin Romberg, Caltech
% Email: jrom@acm.caltech.edu
% Created: October 2005
%
45 function xp = l1eq_pd(x0, A, At, b, pdtol, pdmaxiter, ...
%                               cgtol, cgmaxiter)
%
% largescale = isa(A, 'function_handle');
50 if (nargin < 5), pdtol = 1e–3; end
if (nargin < 6), pdmaxiter = 50; end
if (nargin < 7), cgtol = 1e–8; end
if (nargin < 8), cgmaxiter = 200; end

```

```

55 N = length(x0);

    alpha = 0.01;
    beta = 0.5;
60 mu = 10;

    gradf0 = [zeros(N,1); ones(N,1)];

    % starting point — make sure that it is feasible
65 if (largescale)
    if (norm(A(x0)-b)/norm(b) > cgtol)
        %disp('Starting point infeasible;
        using x0 = At*inv(AAt)*y. ');
        AAt = @(z) A(At(z));
70 [w,cgres,cgiter] = cgsolve(AAt,b,cgtol,cgmaxiter,0);
        if (cgres > 1/2)
            disp('A*At is ill-conditioned:
            .....cannot find starting point');
            xp = x0;
75 return;
        end
        x0 = At(w);
    end
else
80 if (norm(A*x0-b)/norm(b) > cgtol)
        %disp('Starting point infeasible;
        using x0 = At*inv(AAt)*y. ');
        opts.POSDEF = true; opts.SYM = true;
        [w,hcond] = linsolve(A*A', b, opts);
85 if (hcond < 1e-14)
            disp('A*At is ill-conditioned:
            .....cannot find starting point');
            xp = x0;
            return;
90 end
        x0 = A'*w;
    end
end
x = x0;

```

```

95 u = (0.95)*abs(x0) + (0.10)*max(abs(x0));

    % set up for the first iteration
    fu1 = x - u;
    fu2 = -x - u;
100 lamu1 = -1./fu1;
    lamu2 = -1./fu2;
    if (largescale)
        v = -A(lamu1-lamu2);
        Atv = At(v);
105     rpri = A(x) - b;
    else
        v = -A*(lamu1-lamu2);
        Atv = A'*v;
        rpri = A*x - b;
110 end

    sdg = -(fu1'*lamu1 + fu2'*lamu2);
    tau = mu*2*N/sdg;

115 rcent = [-lamu1.*fu1; -lamu2.*fu2] - (1/tau);
    rdual = gradf0 + [lamu1-lamu2; -lamu1-lamu2]
                + [Atv; zeros(N,1)];
    resnorm = norm([rdual; rcent; rpri]);

120 pditer = 0;
    done = (sdg < pdtol) | (pditer >= pdmaxiter);
    while (~done)

        pditer = pditer + 1;
125
        w1 = -1/tau*(-1./fu1 + 1./fu2) - Atv;
        w2 = -1 - 1/tau*(1./fu1 + 1./fu2);
        w3 = -rpri;

130
        sig1 = -lamu1./fu1 - lamu2./fu2;
        sig2 = lamu1./fu1 - lamu2./fu2;
        sigx = sig1 - sig2.^2./sig1;

        if (largescale)

```



```

135     w1p = w3 - A(w1./sigx - w2.*sig2./(sigx.*sig1));
        h11pfun = @(z) -A(1./sigx.*At(z));
        [dv, cgres, cgiter] =
                cgsolve(h11pfun, w1p, cgtol, cgmaxiter, 0);
        if (cgres > 1/2)
140         disp('Cannot solve system.
        ~~~~~Returning previous iterate. ');
            xp = x;
            return
        end
145     dx = (w1 - w2.*sig2./sig1 - At(dv))./sigx;
        Adx = A(dx);
        Atdv = At(dv);
    else
        w1p = -(w3 - A*(w1./sigx - w2.*sig2./(sigx.*sig1)));
150     H11p = A*(sparse(diag(1./sigx))*A');
        opts.POSDEF = true; opts.SYM = true;
        [dv, hcond] = linsolve(H11p, w1p, opts);
        if (hcond < 1e-14)
155         disp('Matrix ill-conditioned.
        ~~~~~Returning previous iterate. ');
            xp = x;
            return
        end
160     dx = (w1 - w2.*sig2./sig1 - A'*dv)./sigx;
        Adx = A*dx;
        Atdv = A'*dv;
    end

    du = (w2 - sig2.*dx)./sig1;

165     dlamu1 = (lamu1./fu1).*(-dx+du) - lamu1 - (1/tau)*1./fu1;
        dlamu2 = (lamu2./fu2).*(dx+du) - lamu2 - 1/tau*1./fu2;

    % make sure that the step is feasible:
170     %keeps lamu1, lamu2 > 0, fu1, fu2 < 0
        indp = find(dlamu1 < 0); indn = find(dlamu2 < 0);
        s = min([1; -lamu1(indp)./dlamu1(indp);
                -lamu2(indn)./dlamu2(indn)]);
        indp = find((dx-du) > 0); indn = find((-dx-du) > 0);

```

```

175  s = (0.99)*min([s; -fu1(indp)./(dx(indp)-du(indp));
                  -fu2(indn)./(-dx(indn)-du(indn))]);

      % backtracking line search
      suffdec = 0;
180  backiter = 0;
      while (~suffdec)
          xp = x + s*dx;  up = u + s*du;
          vp = v + s*dv;  Atvp = Atv + s*Atdv;
          lamu1p = lamu1 + s*dlamu1;  lamu2p = lamu2 + s*dlamu2;
185  fu1p = xp - up;  fu2p = -xp - up;
          rdp = gradf0 + [lamu1p-lamu2p; -lamu1p-lamu2p]
                      + [Atvp; zeros(N,1)];
          rcp = [-lamu1p.*fu1p; -lamu2p.*fu2p] - (1/tau);
          rpp = rpri + s*Adx;
190  suffdec = (norm([rdp; rcp; rpp])<=(1-alpha*s)*resnorm);
          s = beta*s;
          backiter = backiter + 1;
          if (backiter > 32)
              disp('Stuck backtracking, returning last iterate.')
195  xp = x;
              return
          end
      end
      end

200  % next iteration
      x = xp;  u = up;
      v = vp;  Atv = Atvp;
      lamu1 = lamu1p;  lamu2 = lamu2p;
      fu1 = fu1p;  fu2 = fu2p;

205  % surrogate duality gap
      sdg = -(fu1'*lamu1 + fu2'*lamu2);
      tau = mu*2*N/sdg;
      rpri = rpp;
210  rcent = [-lamu1.*fu1; -lamu2.*fu2] - (1/tau);
      rdual = gradf0 + [lamu1-lamu2; -lamu1-lamu2]
                    + [Atv; zeros(N,1)];
      resnorm = norm([rdual; rcent; rpri]);

```

```

215 done = (sdg < pdtol) | (pditer >= pdmaxiter);

    %disp(sprintf('Iteration = %d, tau = %8.3e, ...
    %Primal = %8.3e, PDGap = %8.3e, ...
    %Dual res = %8.3e, Primal res = %8.3e', ...
220 % pditer, tau, sum(u), sdg, norm(rdual), norm(rpri)));
    if (largescale)
        %disp(sprintf('...
        %CG Res = %8.3e, CG Iter = %d', cgres, cgiter));
    else
225 %disp(sprintf('...
        %H11p condition number = %8.3e', hcond));
    end

end
end

```

A.4 Gradients conjugats

```

0 % cgsolve.m
  %
  % Solve a symmetric positive definite system  $Ax = b$  via
  % conjugate gradients.
  %
5 % Usage:
  %  $[x, res, iter] = cgsolve(A, b, tol, maxiter, verbose)$ 
  %
  %  $A$  – Either an  $N \times N$  matrix, or a function handle.
  %
10 %  $b$  –  $N$  vector
  %
  %  $tol$  – Desired precision. Algorithm terminates when
  %  $norm(Ax-b)/norm(b) < tol$  .
  %
15 %  $maxiter$  – Maximum number of iterations.
  %
  %  $verbose$  – If 0, do not print out progress messages.
  % If and integer greater than 0, print out progress
  % every 'verbose' iters.

```

```

20  %
    % Written by: Justin Romberg, Caltech
    % Email: jrom@acm.caltech.edu
    % Created: October 2005
    %
25  function [x,res,iter] = cgsolve(A,b,tol,maxiter,verbose)

    if (nargin < 5), verbose = 1; end

30  implicit = isa(A,'function_handle');

    x = zeros(length(b),1);
    r = b;
    d = r;
35  delta = r'*r;
    delta0 = b'*b;
    numiter = 0;
    bestx = x;
    bestres = sqrt(delta/delta0);
40  while ((numiter < maxiter) && (delta > tol^2*delta0))

        % q = A*d
        if (implicit), q = A(d); else q = A*d; end

45  alpha = delta/(d'*q);
    x = x + alpha*d;

    if (mod(numiter+1,50) == 0)
        % r = b - Aux*x
50  if (implicit), r = b - A(x); else r = b - A*x; end
        else
            r = r - alpha*q;
        end

55  deltaold = delta;
    delta = r'*r;
    beta = delta/deltaold;
    d = r + beta*d;
    numiter = numiter + 1;

```

```
60  if (sqrt(delta/delta0) < bestres)
    bestx = x;
    bestres = sqrt(delta/delta0);
end

65  if ((verbose) && (mod(numiter, verbose)==0))
    disp(sprintf('cg: Iter = %d, Best residual = %8.3e, ...
    Current residual = %8.3e', ...
    numiter, bestres, sqrt(delta/delta0)));
end

70 end

if (verbose)
    disp(sprintf('cg: Iterations = %d, ...
75  Best residual = %14.8e', numiter, bestres));
end
x = bestx;
res = bestres;
iter = numiter;
```


Apèndix B

Codi font: exemple 7.2

B.1 Valors de recuperació en $TV(x)$ (color)

```
0 function [v, Fs, rec, ERROR] = recCLR(PoL, m, M, skip, fix)
  %INPUT
  % PoL - 0 si volem utilitzar la màscara uniforme (punts),
  %       1 si volem utilitzar la màscara lineal (linies)
  % m - nombre mínim de punts o de linies (depenent de PoL)
5 % M - nombre màxim de punts o de linies (depenent de PoL)
  % skip - diferència entre punts/linies a cada recuperació.
  % fix - (màsc. unif) 1 si fixam la màscara per a recuperar
  %       tots els colors de cada recuperació; 0 altrament.
  %OUTPUT
10 %v - valors de punts del mostreig
  % Fs - Error per a cada valor de v.
  % rec - (matriu 4D) Imatges en color recuperades
  % ERROR - (matriu 4D) Error en cada valor de rec.

15 im=im2double(imread('lenna.jpg'));
  [~, n]=size(im(:, :, 1));

  clr = {'_vermell' '_verd' '_blau'};
  quant = M - m;
20 Q = ceil((quant+1)/skip);
  rec = zeros(n, n, 3, Q);
  ERROR = zeros(n, n, 3, Q);
```

```

Fs = zeros(3,Q);
v = zeros(1,Q);
25 counter = 1;

if PoL == 0,
    for j=0:skip:quant,
        if fix==1
30 [M, Mh, mh, mhi]=UniformRandomMask(m+j,n);
        for i=1:3,
            disp(strcat('***IT: ',num2str(j/skip+1),...
                '/' ,num2str(Q),' (Color ', clr(i), ')'));
            actual = im(:,:,i);
35 [~, Xs, ~, ~, ~, ~, ~, ERROR3D]=...
            tveq_phantom_example(actual, 7, n, 111,...
                M, Mh, mh, mhi,0);
            rec(:,:,i,counter)=Xs;
            ERROR(:,:,i,counter)=ERROR3D;
40 Fs(i,counter)=max(sum(ERROR3D));
            parcial=Fs(i,counter)
        end
        v(counter)=m+j;
    else
45 for i=1:3,
        disp(strcat('***IT: ',num2str(j/skip+1),...
            '/' ,num2str(Q),' (Color ', clr(i), ')'));
        actual = im(:,:,i);
        [vars, F, Xs, ERROR3D]=...
50 recMinTV(m+j, m+j, n, 1, actual);
        rec(:,:,i,counter)=Xs;
        ERROR(:,:,i,counter)=ERROR3D;
        Fs(i,counter)=F;
        parcial=Fs(i,counter)
55 end
        v(counter)=vars;
    end
    figure, imshow(rec(:,:,counter));
    counter=counter+1;
60 end
else
    for j=m:skip:M,

```



```

    for i=1:3,
        disp(strcat('***IT: ', num2str(counter), '/ ', ...
65         num2str(Q), ' (Color ', clr(i), ')'));
        actual=im(:,:,i);
        [vars, F, Xs, ERROR3D]=...
        recMinTV_lines(j, j, n, actual);
        rec(:,:,i, counter)=Xs;
70         ERROR(:,:,i, counter)=ERROR3D;
        Fs(i, counter)=F;
        parcial=Fs(i, counter)
    end
    figure, imshow(rec(:,:, :, counter));
75     v(counter)=vars;
        counter=counter+1;
    end
end

```

B.2 Valors de recuperació en $TV(x)$ (uniforme)

```

0 function [vars, F, Xs, ERROR3D]=recMinTV(nk, nK, n, skip, im)
    % Retorna el gràfic de l'error segons el paràmetre K.

    % INPUT:
    % [nk, nK] - interval de presa de mostres
    % n        - Mida de la imatge (N=n·n)
5  % skip     - Distància entre mostres
    % im      - Si està buid s'agafa el fantasma de Sheep-Logan.

    % OUTPUT:
10 % vars    - nombre de mostres preses a cada iteració
    % F      - error en la recuperació segons el valor de K
    % Xs     - matriu tridim., a cada nivell hi ha la recuperació
    % ERROR3D - Matriu tridimensional, a cada nivell
    %         de profunditat hi ha la matriu error (X-Xp).
15 %
    % EXEMPLES:
    % [vars, f, Xs, E3D]=recMinTV(2000, 8000, 256, 200);

```

```

%Inicialitzam les variables i certs valors
20 N = n*n;
   KS = nK-nk;
   corba = zeros(1, ceil((KS+1)/skip));
   ERROR3D = zeros(n,n, ceil((KS+1)/skip));
   Xs=zeros(n,n, ceil((KS+1)/skip));
25
   if nargin > 3,
       phant=im;
   else
       phant=0;
30 end;

   count = 1;
   for i=0:skip:KS,
       disp( ['>>>_CURRENT_ITERATION_>>>' , num2str(i/skip+1), '/' ,
35         num2str(ceil((KS+1)/skip))] );
       %Creació de la màscara amb nk+i punts.
       [M, Mh, mh, mhi] = UniformRandomMask(nk+i, n);
       %Recuperam la imatge
       [X, Xtv, Xbp, M, Mh, mh, mhi, ERROR]=
40 tveq_phantom_example(phant, 7, n, 33, M, Mh, mh, mhi, 0);
       %Guardam els resultats
       ERROR3D(:, :, count) = ERROR;
       Xs(:, :, count) = Xtv;
       corba(count) = sqrt(sum(sum(ERROR.*ERROR)));
45       count = count + 1;
   end

   %Cream les variables i ho representam
   varsK=nk:skip:nK;
50
   figure ,
   plot(varsK, corba);

   vars=varsK;
55 F=corba;

```

B.3 Valors de recuperació en $TV(x)$ (lineal)

```

0 function [vars,F,Xs,ERROR3D]=recMinTV_lines(1, L, n, im)
  % Retorna el gràfic de l'error segons el paràmetre K.

  % INPUT:
  % [l,L] – interval de presa de línies de mostres
  % n – Mida de la imatge (N=n·n)
5  % im – Si està buid s'agafa el fantasma de Sheep–Logan.
  % OUTPUT:
  % vars – nombre de mostres preses a cada iteració
  % F – error en la recuperació segons el valor de K
10 % Xs – matriu tridim., a cada nivell hi ha la recuperació
  % ERROR3D – Matriu tridimensional, a cada nivell
  % de profunditat hi ha la matriu error (X–Xp).

  % EXEMPLES:
15 % [vars, f, Xs, E3D]=recMinTV_lines(2,25,256);

  %Inicialitzam les variables i certs valors
  N=n*n;
  corba = zeros(1,L–l+1);
20 varsL = zeros(1,L–l+1);
  ERROR3D = zeros(n,n,L–l+1);
  Xs = zeros(n,n,L–l+1);
  if nargin>3,
    phant=im;
25 else
    phant=0;
  end;

  for i=l–1:(L–1),
30 disp ([ '>>_CURRENT_ITERATION_>>' ,num2str(i) , '/' ,
  num2str(L–l+1)]);
  %Cream la màscara
  [M, Mh, mh, mhi]=LineMask(i+1,n);
  %Recuperam la imatge
35 [X, Xtv, Xbp, M, Mh, mh, mhi, ERROR]=
  tveq_phantom_example(phant, 7, n, 33, M, Mh, mh, mhi,0);
  %Guardam els resultats

```

```

    ERROR3D(:, :, i+1) = ERROR;
    Xs(:, :, i+1) = Xtv;
40   corba(i+1)=sqrt(sum(sum(ERROR.*ERROR)));
    varsL(i+1) = nnz(M);
    parcial=corba(i+1)
end
45 %Ho representam
    figure ,
    plot(varsL , corba );

vars=varsL ;
50 F=corba ;

```

B.4 Recuperació en $TV(x)$

```

0  % tveq_phantom_example.m
   %
   % Phantom reconstruction from uniform samples
   % in the Fourier plane.
   %
5  % Written by: Justin Romberg, Caltech
   % Email: jrom@acm.caltech.edu
   % Created: October 2005
   % Modified by: Sebastià Pons, 2013

10 function [X, Xtv, Xbp, M, Mh, mh, mhi, ERROR]=
    tveq_phantom_example(phantType, nEllipse, n, PorL, ...
    M, Mh, mh, mhi)
   % INPUT:
   % phantType - 1 (random phantom) / 0 (Sheep-Logan phantom)
15 %             Other (this "other" MUST be a given phantom)
   % nEllipse - Number of ellipses on the random phantom.
   %             If phantomType==1, must be a positive integer.
   % n - dimension of the image (it will be a n*n square).
   % PorL - 1 if we want a uniform mask with points.
20 %             0 if we want a line mask
   % [M,Mh,mh,mhi] - Mask to use. Not used if PorL is 0 or 1.

```

```

%
%OUTPUT:
% X – Phantom used.
25 % [M, Mh, mh, mhi] – Used mask
% Example:
% [X, M, Mh, mh, mhi]=tveq_phantom_example(1,10, 256,1, ...
% [], [], [], []);

30 % Phantom
N = n*n;
if phantType==0,
    X = phantom(n);
else
35     if phantType==1,
        X=randomPhantom(n, nEllipse);
        else
            X=phantType;
        end
40 end;

figure ,
subplot(2,2,1);imshow(X);
x = X(:);
45
L=-1;
if PorL==1,
    % number of points in the Fourier domain
    P = 9200;
50    P=input('m_=_ ');
    % Fourier samples we are given
    [M,Mh,mh,mhi] = UniformRandomMask(P,n);
else
    if PorL==0,
55        % number of lines in the Fourier domain
        L = 22;
        L=input('L_=_ ');
        % Fourier samples we are given
        [M,Mh,mh,mhi] = LineMask(L,n);
60    end
    %else: we use the given one

```

```

end

subplot(2,2,2);imshow(circshift(M,[ceil(n/2) ceil(n/2)]));
65 OMEGA = mhi;
A = @(z) A_fhp(z, OMEGA);
At = @(z) At_fhp(z, OMEGA, n);

% measurements
70 y = A(x);
if L>0, disp(sprintf('m=%5i ', length(y))); end; disp(' ');

% min l2 reconstruction (backprojection)
xbp = At(y);
75 Xbp = reshape(xbp,n,n);
subplot(2,2,3);imshow(Xbp);

% recovery
tic
80 tvI = sum(sum(sqrt([diff(X,1,2) zeros(n,1)].^2 +...
[diff(X,1,1); zeros(1,n)].^2 )));
disp(sprintf('Original_TV=%8.3f ', tvI));
xp = tveq_logbarrier(xbp, A, At, y, 1e-1, 2, 1e-8, 600);
Xtv = reshape(xp, n, n);
85 subplot(2,2,4);imshow(Xtv);
toc
ERROR= abs(Xtv-X);

```

B.5 Creació de la màscara uniforme

```

0 % UniformRandomMask.m
%
% Returns the indicator of the domain in 2D fourier space
% for a uniform random P points.
% Usage : [M,Mh,mi,mhi] = UniformRandomMask(P,N)
5 %
% Written by : Sebastià pons Burguera
% Created : 15/05/2013

```

```

10 function [M,Mh,mi,mhi] = UniformRandomMask(P,N)
M=zeros(N);
x=randint(1,1,[1,N]);
15 y=randint(1,1,[1,N]);
for i=1:P,
    while (M(x,y)==1),
        x=randint(1,1,[1,N]);
        y=randint(1,1,[1,N]);
20    end
    M(x,y)=1;
end

% upper half plane mask (not including origin)
25 Mh = zeros(N);
Mh = M;
Mh(N/2+2:N,:) = 0;
Mh(N/2+1,N/2+1:N) = 0;

30 M = ifftshift(M);
mi = find(M);
Mh = ifftshift(Mh);
mhi = find(Mh);

```

B.6 Creació de la màscara lineal

```

0 % LineMask.m
%
% Returns the indicator of the domain in 2D fourier space
% for the specified line geometry.
% Usage : [M,Mh,mi,mhi] = LineMask(L,N)
5 %
% Written by : Justin Romberg
% Created : 1/26/2004
% Revised : 12/2/2004

```

```

10 function [M,Mh,mi,mhi] = LineMask(L,N)

    thc = linspace(0, pi-pi/L, L);

    %thc = linspace(pi/(2*L), pi-pi/(2*L), L);
15 M = zeros(N);

    % full mask
    for ll = 1:L
20         if ((thc(ll) <= pi/4) | (thc(ll) > 3*pi/4))
                yr = round(tan(thc(ll))*(-N/2+1:N/2-1))+N/2+1;
                for nn = 1:N-1
                M(yr(nn),nn+1) = 1;
                end
25         else
                xc = round(cot(thc(ll))*(-N/2+1:N/2-1))+N/2+1;
                for nn = 1:N-1
                M(nn+1,xc(nn)) = 1;
                end
30         end
    end

    % upper half plane mask
    % (not including origin)
35 Mh = zeros(N);
    Mh = M;
    Mh(N/2+2:N,:) = 0;
    Mh(N/2+1,N/2+1:N) = 0;
40 % Invers of the Fourier and other stuff

    M = ifftshift(M);
    mi = find(M);
45 Mh = ifftshift(Mh);
    mhi = find(Mh);

```


B.7 Constructor de la matriu de mesura

```

0  % A_fhp.m
   %
   % Takes measurements in the upper half-plane of the 2D
   % Fourier transform.
   %
5  % Usage: b = A_fhp(x, OMEGA)
   %
   % x - N vector
   %
   % b - K vector = [mean; real part(OMEGA); imag part(OMEGA)]
10 %
   % OMEGA - K/2-1 vector denoting which Fourier coefficients
   % to use (the real and imag parts of each freq are kept).
   %
   % Written by: Justin Romberg, Caltech
15 % Created: October 2005
   % Email: jrom@acm.caltech.edu
   %

   function y = A_fhp(x, OMEGA)
20
   n = round(sqrt(length(x)));

   yc = 1/n*fft2(reshape(x,n,n));
   y = [yc(1,1); sqrt(2)*real(yc(OMEGA));
25 sqrt(2)*imag(yc(OMEGA))];

```

B.8 Constructor de matriu de mesura adjunta

```

0  % Adjoint of At_fhp (2D Fourier half plane measurements).
   % Usage: x = At_fhp(b, OMEGA, n)
   %
   % b - K vector = [mean; real part(OMEGA); imag part(OMEGA)]
   %
5  % OMEGA - K/2-1 vector denoting which Fourier coefficients
   % to use (the real and imag parts of each freq are kept).

```

```

%
% n – Image is n×n pixels
%
10 % x – N vector
%
% Written by: Justin Romberg, Caltech
% Created: October 2005
% Email: jrom@acm.caltech.edu
15
function x = At_fhp(y, OMEGA, n)

K = length(y);
fx = zeros(n,n);
20 fx(1,1) = y(1);
fx(OMEGA) = sqrt(2)*(y(2:(K+1)/2) + i*y((K+3)/2:K));
x = reshape(real(n*ifft2(fx)), n*n, 1);

```

B.9 Solucionador de programació lineal (log-barrera)

```

0 % tveq_logbarrier.m
%
% Solve equality constrained TV minimization
% min TV(x) s.t. Ax=b.
%
5 % Recast as the SOCP
% min sum(t) s.t. ||D_{ij}x||_2 <= t, i,j=1,...,n
% Ax=b
% and use a log barrier algorithm.
%
10 % Usage: xp = tveq_logbarrier(x0, A, At, b, lbtol, ...
% mu, slqtol, slqmaxiter)
%
% x0 – N×1 vector, initial point.
%
15 % A – Either a handle to a function that takes a N vector
% and returns a K vector, or a K×N matrix. If A is a
% function handle, the algorithm operates in
% "largescale" mode, solving the Newton systems via

```

```

%      the Conjugate Gradients algorithm.
20 %
% At – Handle to a function that takes a K vector and
%      returns an N vector. If A is a KxN matrix, At is
%      ignored.
%
25 % b – Kx1 vector of observations.
%
% lbtol – The log barrier algorithm terminates when the
%          duality gap <= lbtol. Also, the number of log
%          barrier iterations is completely determined by
30 %          lbtol. Default = 1e-3.
%
% mu – Factor by which to increase the barrier constant at
%       each iteration. Default = 10.
%
35 % slqtol – Tolerance for SYMMLQ; ignored if A is a matrix.
%           Default = 1e-8.
%
% slqmaxiter – Maximum number of iterations for SYMMLQ;
%              ignored if A is a matrix.
40 %           Default = 200.
%
% Written by: Justin Romberg, Caltech
% Email: jrom@acm.caltech.edu
% Created: October 2005
45 %

function xp = ...
tveq_logbarrier(x0,A,At,b,lbtol ,mu,slqtol ,slqmaxiter)

50 largescale = isa(A,'function_handle');

if (nargin < 5), lbtol = 1e-3; end
if (nargin < 6), mu = 10; end
if (nargin < 7), slqtol = 1e-8; end
55 if (nargin < 8), slqmaxiter = 200; end

newtontol = lbtol;
newtonmaxiter = 50;

```

```

60 N = length(x0);
   n = round(sqrt(N));

   % create (sparse) differencing matrices for TV
   Dv = spdiags([reshape([-ones(n-1,n); zeros(1,n)],N,1) ...
65   reshape([zeros(1,n); ones(n-1,n)],N,1)], [0 1], N, N);
   Dh = spdiags([reshape([-ones(n,n-1) zeros(n,1)],N,1) ...
   reshape([zeros(n,1) ones(n,n-1)],N,1)], [0 n], N, N);

   % starting point — make sure that it is feasible
70 if (largescale)
   if (norm(A(x0)-b)/norm(b) > slqtol)
       disp('Starting point infeasible; ...
using x0 = At*inv(AAt)*y. ');
       AAt = @(z) A(At(z));
75   [w,cgres] = cgsolve(AAt, b, slqtol, slqmaxiter, 0);
       if (cgres > 1/2)
           disp('A*At is ill-conditioned; ...
cannot find starting point ');
           xp = x0;
80       return;
       end
       x0 = At(w);
   end
else
85   if (norm(A*x0-b)/norm(b) > slqtol)
       disp('Starting point infeasible; ...
using x0 = At*inv(AAt)*y. ');
       opts.POSDEF = true; opts.SYM = true;
       [w, hcond] = linsolve(A*A', b, opts);
90   if (hcond < 1e-14)
       disp('A*At is ill-conditioned; ...
cannot find starting point ');
       xp = x0;
       return;
95   end
       x0 = A'*w;
   end
end
end

```

```

x = x0;
100 Dhx = Dh*x;  Dvx = Dv*x;
    t = (0.95)*sqrt(Dhx.^2 + Dvx.^2) + ...
        (0.1)*max(sqrt(Dhx.^2 + Dvx.^2));

% choose initial value of tau so that the duality gap
105 % after the first step will be about the original TV
    tau = N/sum(sqrt(Dhx.^2+Dvx.^2));

lbiter = ceil((log(N)-log(lbtol)-log(tau))/log(mu));
disp(sprintf('Number of log barrier iterations = %d\n',
110 lbiter));
totaliter = 0;
for ii = 1:lbiter

    [xp, tp, ntiter] =
115     tveq_newton(x, t, A, At, b, tau, newtontol, ...
        newtonmaxiter, slqtol, slqmaxiter);
    totaliter = totaliter + ntiter;

    tvxp = sum(sqrt((Dh*xp).^2 + (Dv*xp).^2));
120     disp(sprintf('\nLog barrier iter = %d, TV = %.3f,
functional = %8.3f, tau = %8.3e, total newton iter = ...
%d\n', ii, tvxp, sum(tp), tau, totaliter));

    x = xp;
125     t = tp;

    tau = mu*tau;

end

```

B.10 Newton per a $TV(x)$

```

0 % tveq_newton.m
%
% Newton algorithm for log-barrier subproblems for TV
% minimization with equality constraints.

```

```

%
5 % Usage:
% [xp, tp, niter] = tveq_newton(x0, t0, A, At, b, tau, ...
%   newtontol, newtonmaxiter, slqtol, slqmaxiter)
%
% x0, t0 - starting points
10 %
% A - Either a handle to a function that takes a N vector
%   and returns a K vector , or a KxN matrix. If A is a
%   function handle, the algorithm operates in
%   "largescale" mode, solving the Newton systems
15 %   via the Conjugate Gradients algorithm.
%
% At - Handle to a function that takes a K vector and
%   returns an N vector. If A is a KxN matrix, At is ignored.
%
20 % b - Kx1 vector of observations.
%
% tau - Log barrier parameter.
%
% newtontol - Terminate when the Newton decrement
25 %               is <= newtontol.
%
% newtonmaxiter - Maximum number of iterations.
%
% slqtol - Tolerance for SYMMLQ; ignored if A is a matrix.
30 %
% slqmaxiter - Maximum number of iterations for SYMMLQ;
%   ignored if A is a matrix.
%
% Written by: Justin Romberg, Caltech
35 % Email: jrom@acm.caltech.edu
% Created: October 2005

function [xp, tp, niter] = tveq_newton(x0, t0, A, At, b, ...
40 tau, newtontol, newtonmaxiter, slqtol, slqmaxiter)

largescale = isa(A, 'function_handle');

alpha = 0.01;

```

```

45  beta = 0.5;
N = length(x0);
n = round(sqrt(N));
K = length(b);

50  % create (sparse) differencing matrices for TV
Dv = spdiags([reshape([-ones(n-1,n); zeros(1,n)],N,1) ...
             reshape([zeros(1,n); ones(n-1,n)],N,1)], [0 1], N, N);
Dh = spdiags([reshape([-ones(n,n-1) zeros(n,1)],N,1) ...
             reshape([zeros(n,1) ones(n,n-1)],N,1)], [0 n], N, N);

55  % auxillary matrices for preconditioning
Mdv = spdiags([reshape([ones(n-1,n); zeros(1,n)],N,1) ...
              reshape([zeros(1,n); ones(n-1,n)],N,1)], [0 1], N, N);
Mdh = spdiags([reshape([ones(n,n-1) zeros(n,1)],N,1) ...
              reshape([zeros(n,1) ones(n,n-1)],N,1)], [0 n], N, N);
60  Mmd=reshape([ones(n-1,n-1) zeros(n-1,1); zeros(1,n)],N,1);

% initial point
x = x0;
65  t = t0;
Dhx = Dh*x;  Dvx = Dv*x;
ft = 1/2*(Dhx.^2 + Dvx.^2 - t.^2);
f = sum(t) - (1/tau)*(sum(log(-ft)));

70  niter = 0;
done = 0;
while (~done)

    ntgx = Dh'*((1./ft).*Dhx) + Dv'*((1./ft).*Dvx);
75  ntgt = -tau - t./ft;
    gradf = -(1/tau)*[ntgx; ntgt];

    sig22 = 1./ft + (t.^2)./(ft.^2);
    sig12 = -t./ft.^2;
80  sigb = 1./ft.^2 - (sig12.^2)./sig22;

    wlp = ntgx - Dh'*(Dhx.*(sig12./sig22).*ntgt)
          - Dv'*(Dvx.*(sig12./sig22).*ntgt);

```

```

wp = [w1p; zeros(K,1)];
85  if (largescale)
    % diagonal of H11p
    dg11p = Mdh*(-1./ft + sigb.*Dhx.^2)
           + Mdv*(-1./ft + sigb.*Dvx.^2)
           + 2*Mmd.*sigb.*Dhx.*Dvx;
90  afac = max(dg11p);
    hpfun = @(z) Hpeval(z, A, At, Dh, Dv, Dhx, Dvx, ...
                       sigb, ft, afac);
    [dxv, slqflag, slqres, slqiter] = symmlq(hpfun, wp, ...
                                             slqtol, slqmaxiter);
95  if (slqres > 1/2)
    disp('Cannot solve system. Returning previous iter. ');
    xp = x;
    return
    end
100 else
    H11p = Dh'*sparse(diag(-1./ft + sigb.*Dhx.^2))*Dh + ...
          Dv'*sparse(diag(-1./ft + sigb.*Dvx.^2))*Dv + ...
          Dh'*sparse(diag(sigb.*Dhx.*Dvx))*Dv + ...
          Dv'*sparse(diag(sigb.*Dhx.*Dvx))*Dh;
105  afac = max(diag(H11p));
    Hp = full([H11p afac*A'; afac*A zeros(K)]);
    %keyboard
    opts.SYM = true;
    [dxv, hcond] = linsolve(Hp, wp, opts);
110  if (hcond < 1e-14)
    disp('Matrix ill-cond. Returning prev. iter. ');
    xp = x; tp = t;
    return
    end
115  end

dx = dxv(1:N);
Dhdx = Dh*dx; Dvdx = Dv*dx;
dt = (1./sig22).*(ntgt - sig12.*(Dhx.*Dhdx + Dvx.*Dvdx));
120  % minimum step size that stays in the interior
    aqt = Dhdx.^2 + Dvdx.^2 - dt.^2;
    bqt = 2*(Dhdx.*Dhx + Dvdx.*Dvx - t.*dt);

```



```

125   cqt = Dhx.^2 + Dvx.^2 - t.^2;

   tsols = [(-bqt+sqrt(bqt.^2-4*aqt.*cqt))./(2*aqt); ...
            (-bqt-sqrt(bqt.^2-4*aqt.*cqt))./(2*aqt) ];
   indt = find([(bqt.^2 > 4*aqt.*cqt);
               (bqt.^2 > 4*aqt.*cqt)] & (tsols > 0));
130   smax = min(1, min(tsols(indt)));
   s = (0.99)*smax;

   % line search
135   suffdec = 0;
   backiter = 0;

   while (~suffdec)
140     xp = x + s*dx;  tp = t + s*dt;
     Dhxp = Dhx + s*Dhdx;  Dvxp = Dvx + s*Dvdx;
     ftp = 1/2*(Dhxp.^2 + Dvxp.^2 - tp.^2);
     fp = sum(tp) - (1/tau)*(sum(log(-ftp)));
     flin = f + alpha*s*(gradf'*[dx; dt]);
     suffdec = (fp <= flin);
145     s = beta*s;
     backiter = backiter + 1;
     if (backiter > 32)
         disp('Stuck_backtracking, returning last iterate. ');
         xp = x;  tp = t;
150     return
     end
   end

   % set up for next iteration
155   x = xp;  t = tp;
   Dvx = Dvxp;  Dhx = Dhxp;
   ft = ftp;  f = fp;

   lambda2 = -(gradf'*[dx; dt]);
160   stepsize = s*norm([dx; dt]);
   niter = niter + 1;
   done=(lambda2/2 < newtontol) | (niter >= newtonmaxiter);

```

```

    disp(sprintf('Newton_iter = %d, Functional = %8.3f, ...
165  Newton_decrement = %8.3f, Stepsize = %8.3e', ...
        niter, f, lambda2/2, stepsize));
    if (largescale)
        disp(sprintf('.....SYMMMLQ_Res = %8.3e, ...
.....SYMMMLQ_Iter = %d', slqres, slqiter));
170  else
        disp(sprintf('.....H11p_condition ...
.....number = %8.3e', hcond));
    end
end
175 % Implicit application of Hessian
function y = Hpeval(z,A,At,Dh,Dv,Dhx,Dvx, sigb,ft,afac)

N = length(ft);
180 K = length(z)-N;
w = z(1:N);
v = z(N+1:N+K);

Dhw = Dh*w;
185 Dvw = Dv*w;

y1 = Dh'*((-1./ft + sigb.*Dhx.^2).*Dhw + ...
sigb.*Dhx.*Dvx.*Dvw) + Dv'*((-1./ft + ...
sigb.*Dvx.^2).*Dvw+sigb.*Dhx.*Dvx.*Dhw)+afac*At(v);
190 y2 = afac*A(w);

y = [y1; y2];

```

B.11 Imatge aleatòria. Imitació “phantom”.

```

0  function img = randomPhantom(dim,n)

    img=zeros(dim);
    %imshow(img);

5  for i=1:n,

```

```

    %y0=randint(1,1,[2,dim-1]);
    y0 = floor(normrnd(dim/2,dim/2,[1]));
    f (y0<0)||(y0>dim), y0 = floor(dim/2);end;
    %x0=randint(1,1,[2,dim-1]);
10    x0 = floor(normrnd(dim/2,dim/2,[1]));
    if (x0<0)||(x0>dim), x0 = floor(dim/2);end;
    m=min(min(y0,x0),min(dim-y0,dim-x0));
    b=randint(1,1,[ceil(m*2/3),m]);
    %b = floor(normrnd(dim/2,dim/2,[1]));
15    %if (b<0)||(b>dim), b = floor(dim/3);end;
    a=randint(1,1,[ceil(b/2),b]);
    teta=rand(1,1)*360;
    color=rand(1,1);
    img=ellipseMatrix(y0, x0, a, b, teta, img, color);
20 end

```

B.12 Representar el·lipse a base de píxels

```

0 function ret = ...
    ellipseMatrix(y0, x0, a, b, teta, Image, color)
    % Set the elements of the Matrix Image which are in the
    % interior of the ellipse E with the value 'color'. The
    % ellipse E has center (y0, x0), the major axe = a, the
5 % minor axe = b, and teta is the angle macked by the
    % major axe with the orizontal axe.
    %
    % Function: ellipseMatrix
    % Version: 1.1
10 % Author: Nicolae Cindea
    % Modified: Sebastià Pons, 2013

    im = Image;
    colored = 0*Image;
15 [ny, nx] = size(im);
    imtemp = zeros(ny, nx);
    list = zeros(ny * nx, 2);
    toplist = 1;
    c = sqrt(a * a - b * b);

```

```

20  x0 = round(x0);
    y0 = round(y0);
    list (toplist , 1) = y0;
    list (toplist , 2) = x0;
25  im(y0, x0) = im(y0, x0)+ color;
    colored(y0, x0)=1;
    if im(y0, x0)>1,
        im(y0, x0)=im(y0, x0)-1;
    end;
30
    while (toplist > 0)
        y = list (toplist , 1);
        x = list (toplist , 2);
        toplist = toplist - 1;
35
        if local_isValid(y, x + 1, y0, x0, a, c, teta, ...
            imtemp, ny, nx, colored)
            toplist = toplist + 1;
            list (toplist , 1) = y;
40            list (toplist , 2) = x + 1;
            im(list (toplist , 1), list (toplist , 2)) =...
            im(list (toplist , 1), list (toplist , 2))+color;
            imtemp(list (toplist , 1), list (toplist , 2)) = ...
            im(list (toplist , 1), list (toplist , 2))+color;
45            if im(list (toplist , 1), list (toplist , 2))>1,
                im(list (toplist , 1), list (toplist , 2))= ...
                im(list (toplist , 1), list (toplist , 2))-1;
            end;
            if imtemp(list (toplist , 1), list (toplist , 2))>1,
50                imtemp(list (toplist , 1), list (toplist , 2))=...
                imtemp(list (toplist , 1), list (toplist , 2))-1;
            end;
            colored(list (toplist , 1), list (toplist , 2))=1;
        end
55    if local_isValid(y - 1, x, y0, x0, a, c, teta, ...
        imtemp, ny, nx, colored)
        toplist = toplist + 1;
        list (toplist , 1) = y - 1;
        list (toplist , 2) = x;

```

```

60     im(list(toplist, 1), list(toplist, 2)) =...
        im(list(toplist, 1), list(toplist, 2))+color;
        imtemp(list(toplist, 1), list(toplist, 2)) =...
        im(list(toplist, 1), list(toplist, 2))+color;
65     if im(list(toplist, 1), list(toplist, 2))>1,
            im(list(toplist, 1), list(toplist, 2))=...
            im(list(toplist, 1), list(toplist, 2))-1;
        end;
        if imtemp(list(toplist, 1), list(toplist, 2))>1,
            imtemp(list(toplist, 1), list(toplist, 2))=...
70     imtemp(list(toplist, 1), list(toplist, 2))-1;
        end;
        colored(list(toplist, 1), list(toplist, 2))=1;
    end
    if local_isValid(y, x - 1, y0, x0, a, c, teta, ...
75         imtemp, ny, nx, colored)
        toplist = toplist + 1;
        list(toplist, 1) = y;
        list(toplist, 2) = x - 1;
        im(list(toplist, 1), list(toplist, 2)) =...
80     im(list(toplist, 1), list(toplist, 2))+color;
        imtemp(list(toplist, 1), list(toplist, 2)) =...
        im(list(toplist, 1), list(toplist, 2))+color;
        if im(list(toplist, 1), list(toplist, 2))>1,
            im(list(toplist, 1), list(toplist, 2))=...
85     im(list(toplist, 1), list(toplist, 2))-1;
        end;
        if imtemp(list(toplist, 1), list(toplist, 2))>1,
            imtemp(list(toplist, 1), list(toplist, 2))=...
90     imtemp(list(toplist, 1), list(toplist, 2))-1;
        end;
        colored(list(toplist, 1), list(toplist, 2))=1;
    end
    if local_isValid(y + 1, x, y0, x0, a, c, teta, ...
95         imtemp, ny, nx, colored) == 1
        toplist = toplist + 1;
        list(toplist, 1) = y + 1;
        list(toplist, 2) = x;
        im(list(toplist, 1), list(toplist, 2)) =...
        im(list(toplist, 1), list(toplist, 2))+color;

```

```

100     imtemp(list(toplist, 1), list(toplist, 2)) =...
        im(list(toplist, 1), list(toplist, 2))+color;
        if im(list(toplist, 1), list(toplist, 2))>1,
            im(list(toplist, 1), list(toplist, 2))=...
            im(list(toplist, 1), list(toplist, 2))-1;
105     end;
        if imtemp(list(toplist, 1), list(toplist, 2))>1,
            imtemp(list(toplist, 1), list(toplist, 2))=...
            imtemp(list(toplist, 1), list(toplist, 2))-1;
        end;
110     colored(list(toplist, 1), list(toplist, 2))=1;
    end
end
ret = im;

```

B.13 Subprograma de “ellipseMatrix”

```

0  function is_val = local_isValid(y, x, y0, x0, a, c, ...
        teta, im, ny, nx, colored)
    [n m]=size(im);
    d1=(x - x0 - c * cos(teta))^2 + (y - y0 - c * sin(teta))^2;
    d1=sqrt(d1);
5  d2=(x - x0 + c * cos(teta))^2 + (y - y0 + c * sin(teta))^2;
    d2=sqrt(d2);
    if (x>0) && (y>0) && (x <= nx) && (y <= ny)&&...
        (d1 + d2 <= 2*a)&& (colored(y, x) ~ = 1)
        is_val = 1;
10  else
        is_val = 0;
    end
end

```

Apèndix C

Figures de l'exemple 7.2

A la figura C.1 podem trobar les imatges que prendrem de mostra per als exemples de recuperació de senyals bidimensionals.

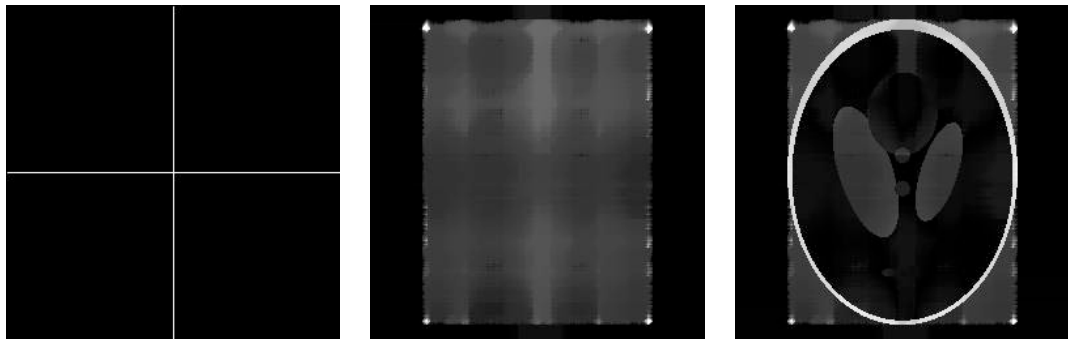


(a) Imatge de mostra. Fantasma de Shepp-Logan (b) Imatge de mostra. Fotografia de na Lenna.

Figura C.1: Imatges de mostra.

Tot seguit, de la figura C.2 a la C.8 hi ha les imatges de la màscara lineal presa, la imatge recuperada i l'error comès per a recuperacions per diferents quantitats de línies. Després, de la figura C.9 a la figura C.16 hi ha el mateix, referent a la màscara uniforme, per a diferents quantitats de punts presos pel mostreig.

Després, en les figures C.17, C.18 i C.19, hi ha els resultats de les recuperacions de les senyals en color utilitzant la màscara uniforme, la màscara uniforme fixant-la per a totes les capes de cada recuperació i la màscara lineal. En cada figura trobareu, per columnes, la imatge a color, la capa roja, la capa verda i la capa blava; i, per files, la recuperació per a certs valors de punts o línies presos.

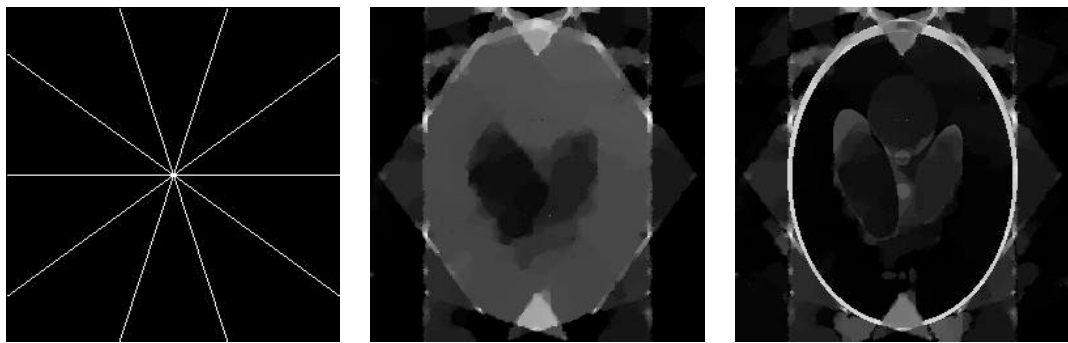


(a) Màscara, 2 línies.

(b) Imatge recuperada.

(c) Error.

Figura C.2: Resultat de la recuperació amb la màscara lineal consistent en 2 línies (509 punts, un 0.7% del total).

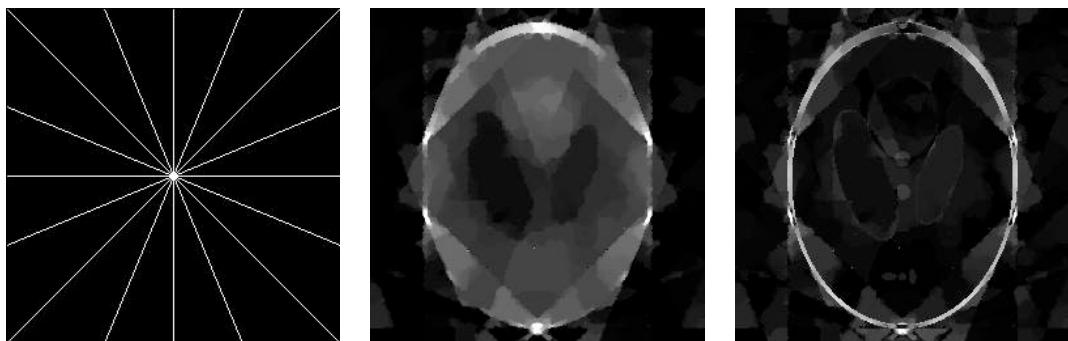


(a) Màscara, 5 línies.

(b) Imatge recuperada.

(c) Error.

Figura C.3: Resultat de la recuperació amb la màscara lineal consistent en 5 línies (1269 punts, un 1.9% del total).



(a) Màscara, 8 línies.

(b) Imatge recuperada.

(c) Error.

Figura C.4: Resultat de la recuperació amb la màscara lineal consistent en 8 línies (2025 punts, un 3% del total).

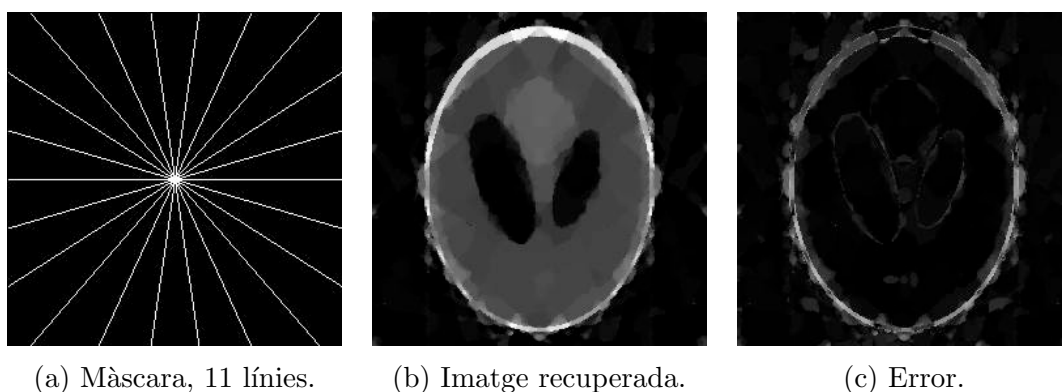


Figura C.5: Resultat de la recuperació amb la màscara lineal consistent en 11 línies (2773 punts, un 4.3% del total).

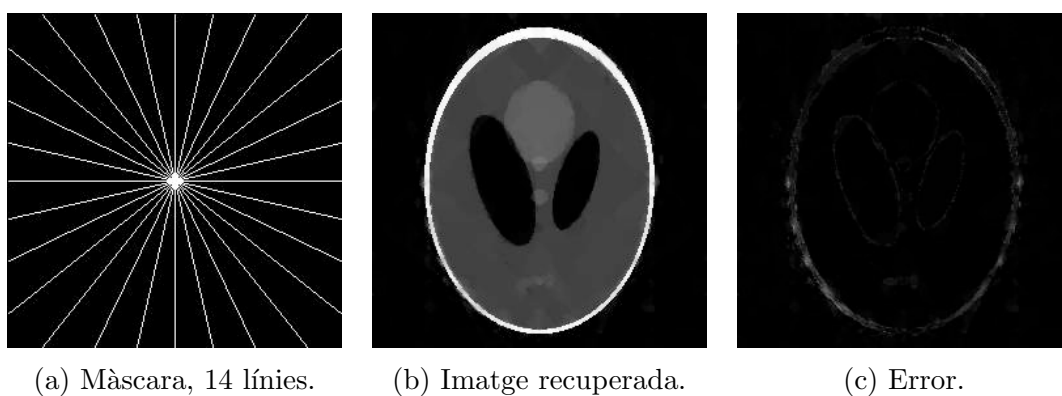


Figura C.6: Resultat de la recuperació amb la màscara lineal consistent en 14 línies (3517 punts, un 5.4% del total).

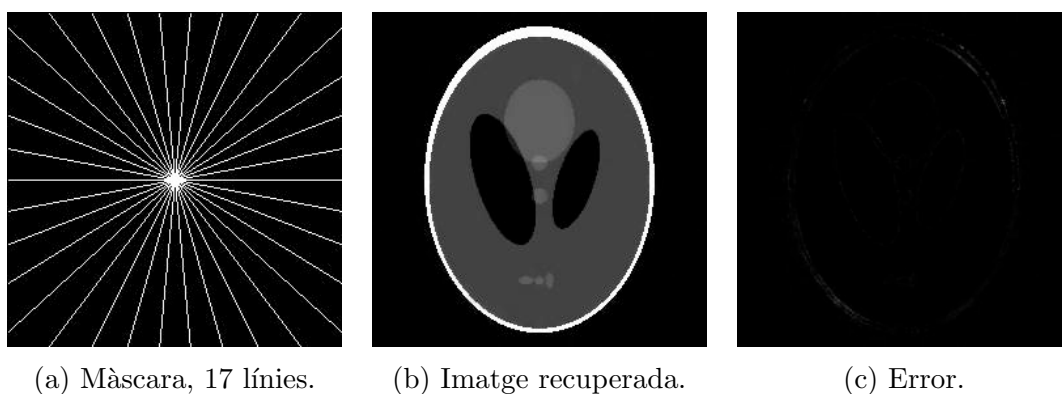


Figura C.7: Resultat de la recuperació amb la màscara lineal consistent en 17 línies (4257 punts, un 6.5% del total).

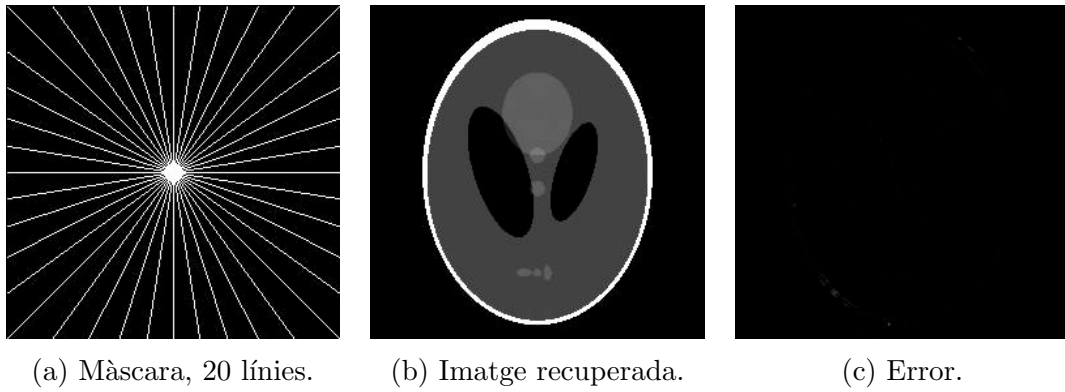


Figura C.8: Resultat de la recuperació amb la màscara lineal consistent en 20 línies (5001 punts, un 7.6% del total).

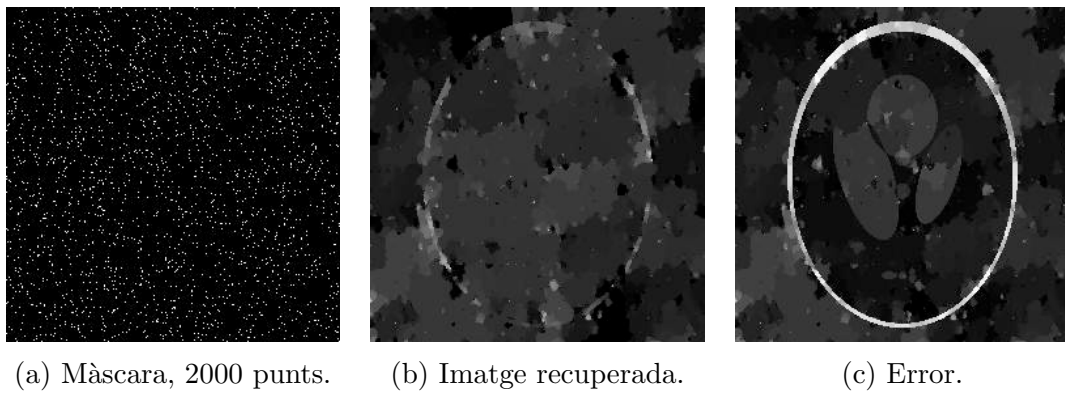


Figura C.9: Resultat de la recuperació amb la màscara uniforme consistent en 2000 punts, un 3.1% del total.

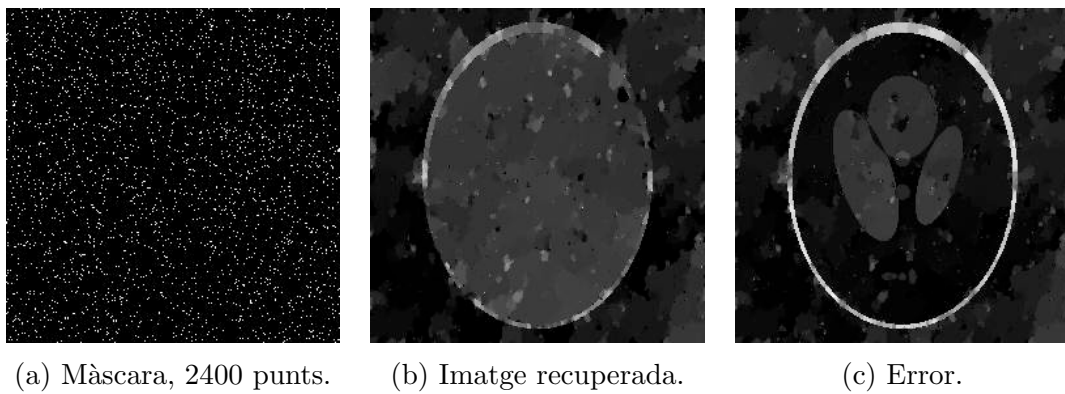


Figura C.10: Resultat de la recuperació amb la màscara uniforme consistent en 2400 punts, un 3.7% del total.

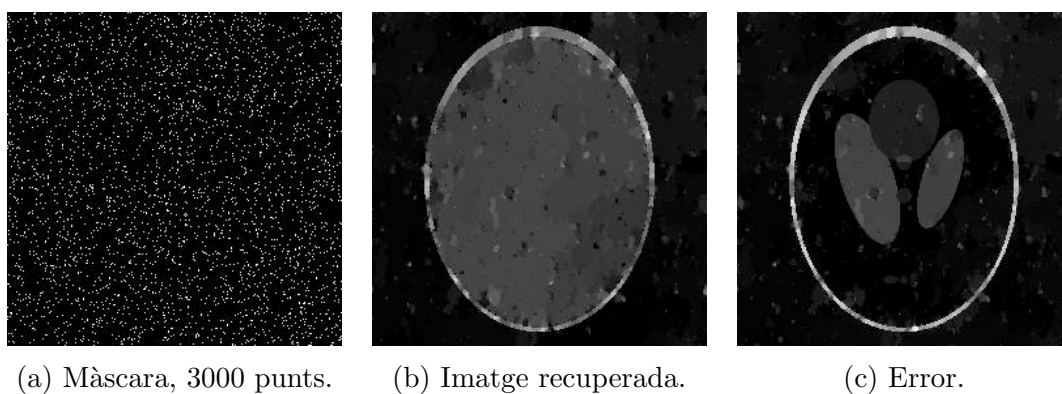


Figura C.11: Resultat de la recuperació amb la màscara uniforme consistent en 3000 punts, un 4.6% del total.

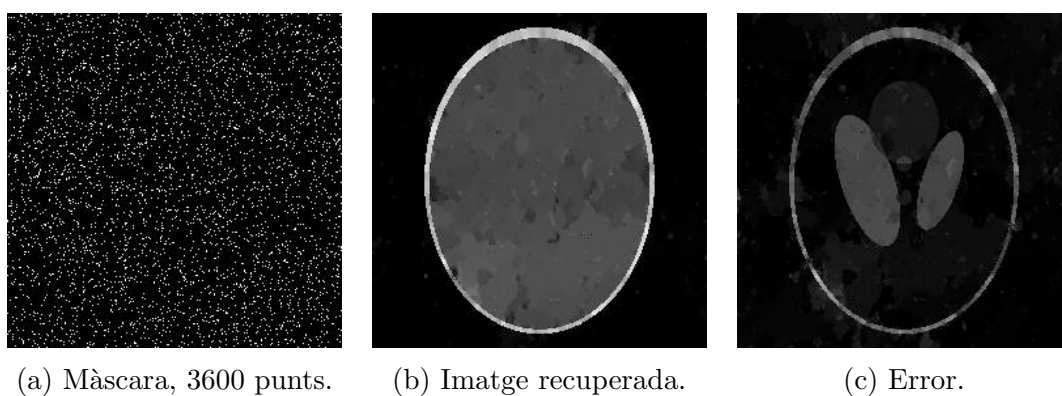


Figura C.12: Resultat de la recuperació amb la màscara uniforme consistent en 3600 punts, un 5.5% del total.

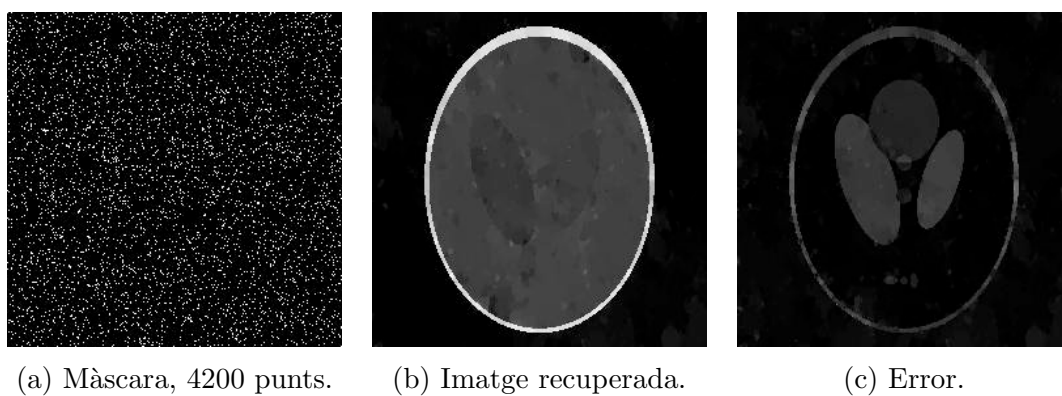


Figura C.13: Resultat de la recuperació amb la màscara uniforme consistent en 4200 punts, un 6.4% del total.

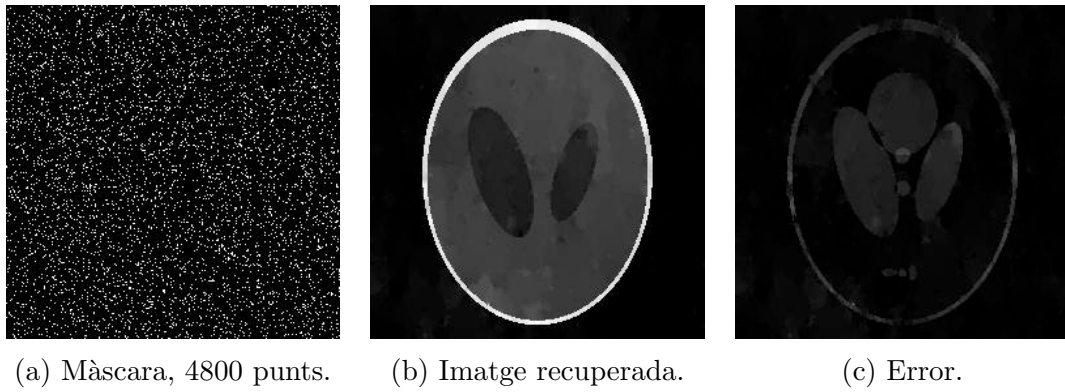


Figura C.14: Resultat de la recuperació amb la màscara uniforme consistent en 4800 punts, un 7.3% del total.

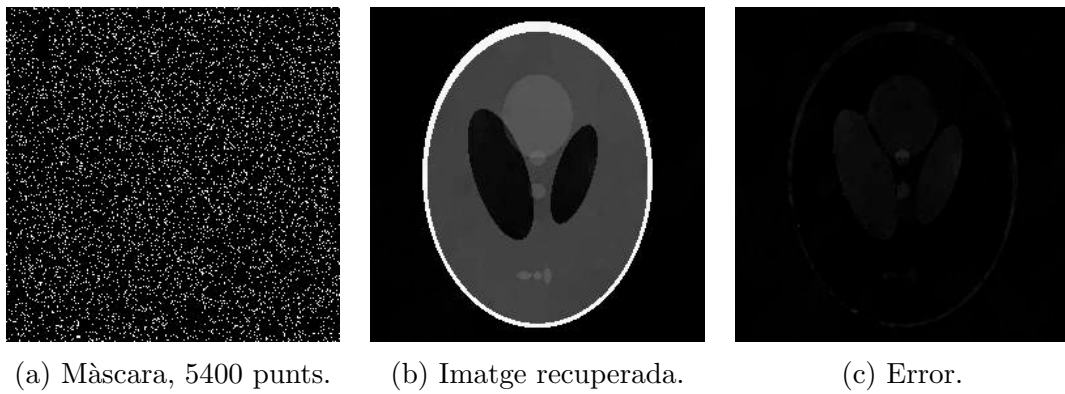


Figura C.15: Resultat de la recuperació amb la màscara uniforme consistent en 5400 punts, un 8.2% del total.

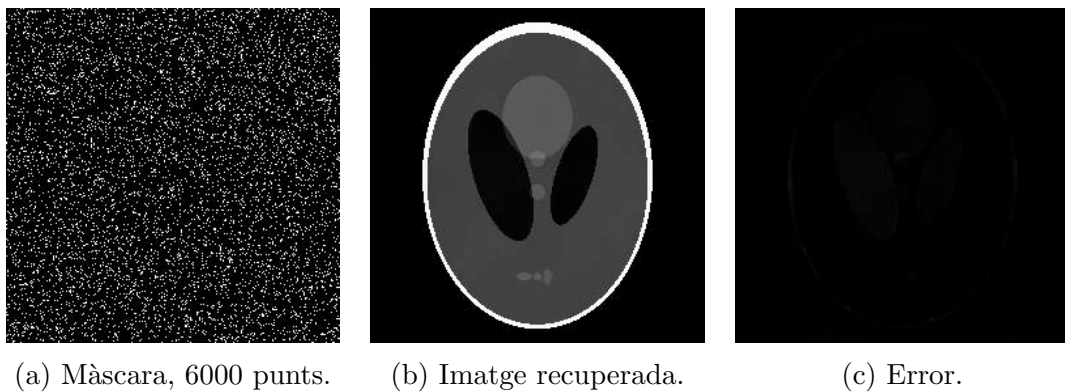


Figura C.16: Resultat de la recuperació amb la màscara uniforme consistent en 6000 punts, un 9.2% del total.

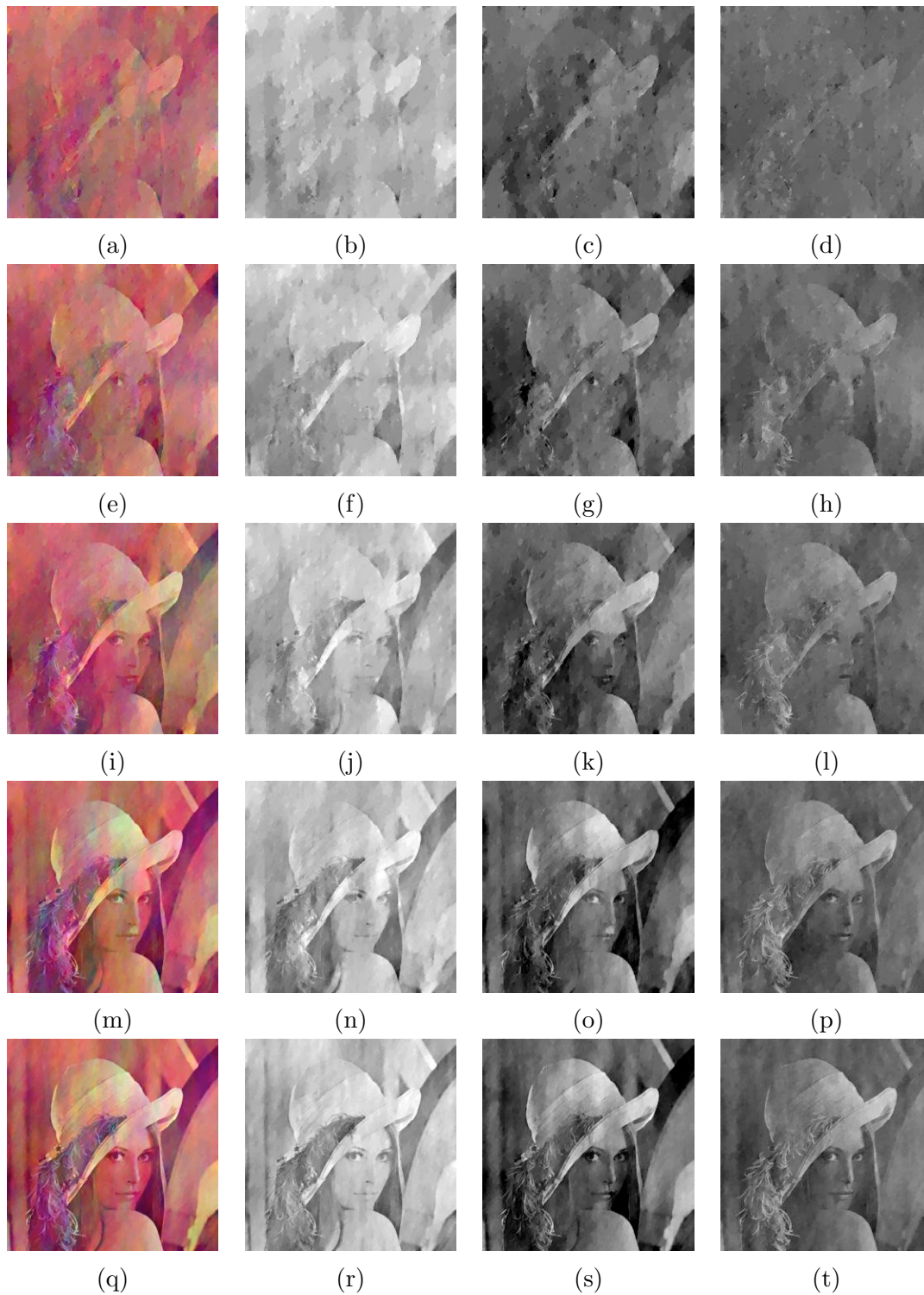


Figura C.17: Recuperació amb la màscara uniforme variant la màscara. Per files correspon a prendre 6000, 10000, 14000, 22000 i 30000 punts, un 9.2%, 15.3%, 21.4%, 33.6% i 45.8% del total de punts. Per columnes, correspon a la imatge en color, a la capa vermella, verda i blava.

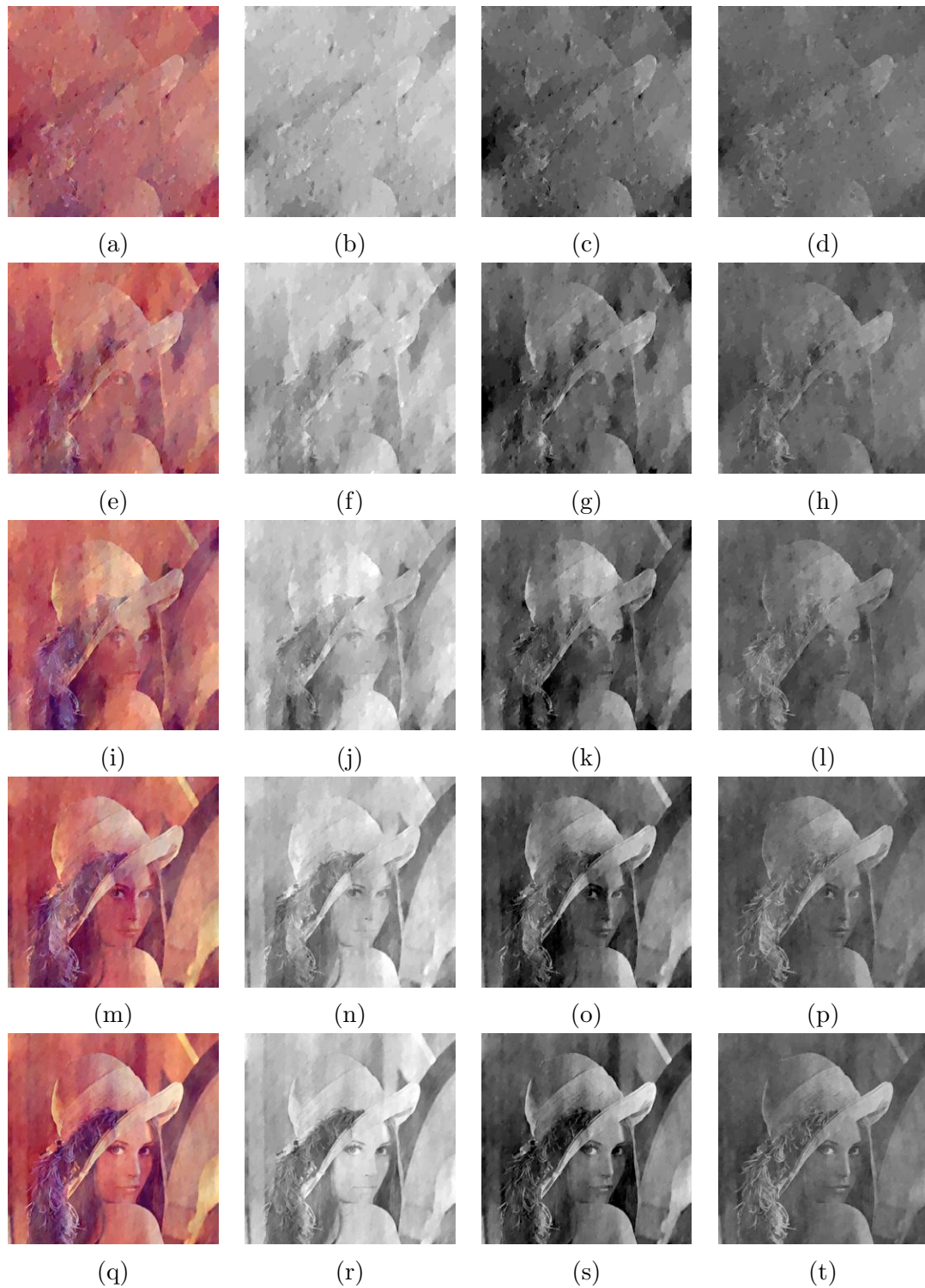


Figura C.18: Recuperació amb la màscara uniforme fixant la màscara per a cada fila. Per files correspon a prendre 6000, 10000, 14000, 22000 i 30000 punts, un 9.2%, 15.3%, 21.4%, 33.6% i 45.8% del total de punts. Per columnes, correspon a la imatge en color, a la capa vermella, verda i blava.

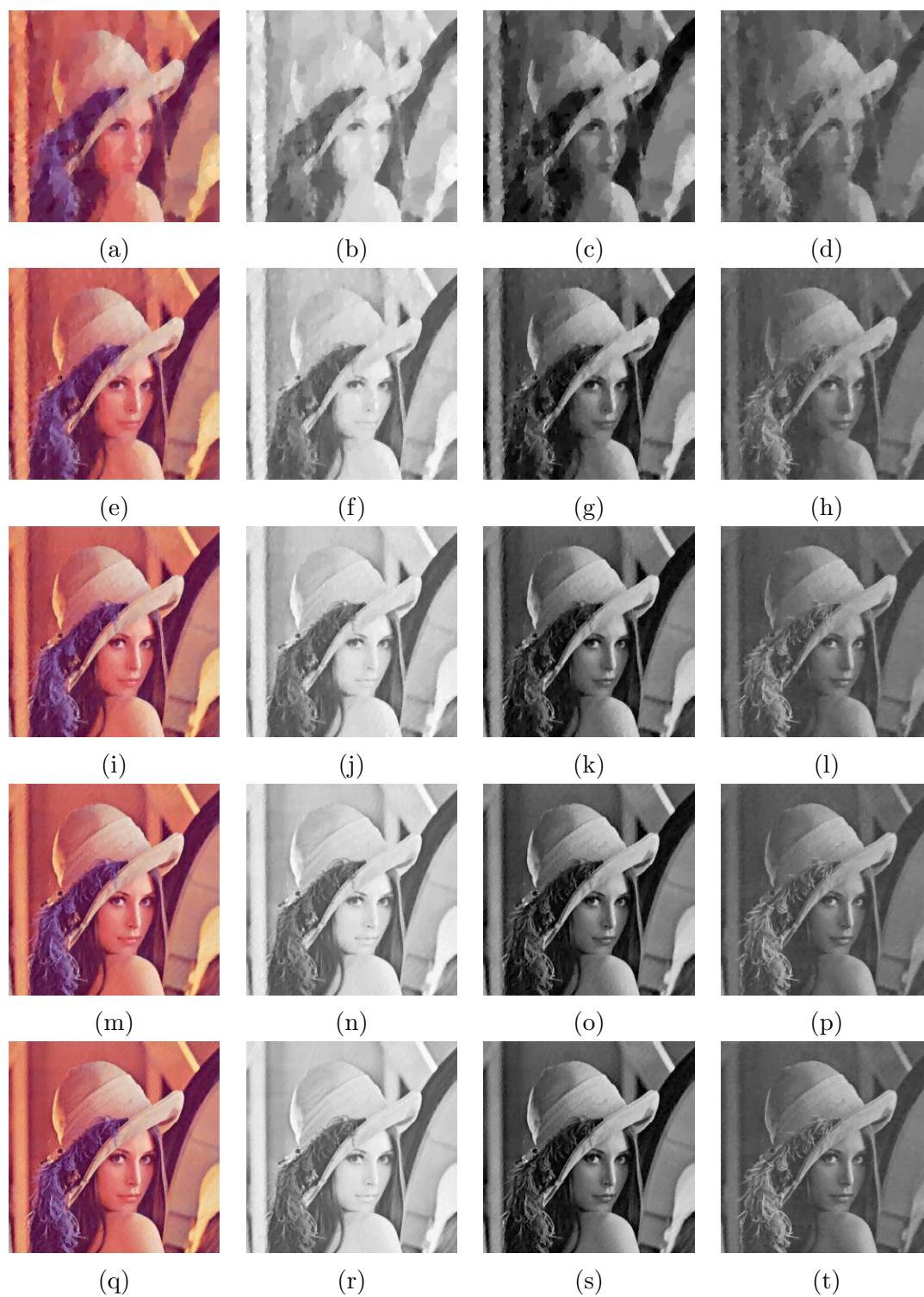


Figura C.19: Recuperació amb la màscara lineal. Per files correspon a prendre 20, 40, 60, 80 i 100 línies, un 7.6%, 14.9%, 21.9%, 28.6% i 35% del total de punts. Per columnes, correspon a la imatge en color, a la capa vermella, verda i blava.

Bibliografia

- [Bou00] S. Bouchero, G. Lugosi i P. Massart, *A sharp concentration inequality with applications*, Random Structures Algorithms, No. 16, pp 277-292, 2000.
- [Boy04] S. Boyd i L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [Can04] E. Candès i J. Romberg, *Recovery of sparse signals via convex programming*, ℓ_1 -MAGIC (<http://users.ece.gatech.edu/~justin/l1magic/>), 2004.
- [Can04b] E. Candès, J. Romberg i T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, Vol. 12, No. 2, pp 489-509, juny 2004.
- [Can04] E. Candès i T. Tao, *Near optimal signal recovery from random projections: Universal encoding strategies?*, IEEE. Trans. Info. Theory, Vol. 52, No. 12, pp 5406-5425, octubre 2004.
- [Can05] E. Candès i T. Tao, *Decoding by linear programming*, IEEE. Trans. Info. Theory, Vol. 51, pp 4203-4215, 2005.
- [Can08] E. Candès i T. Tao, *Reflections on Compressed sensing*, IEEE Information theory society newsletter, desembre 2008.
- [Coi01] R. Coifman, F. Geshwind i Y. Meyer, *Noiselets*, Applied and Computational Harmonic Analysis, Vol.10, pp 27-44, 2001.
- [Cra37] H. Cramer, *Random Variables and Probability Distributions*, Cambridge Tracts in Mathematics and Mathematical Physics, No. 36, pp 8-121, London, Cambridge University Press, 1937.
- [Don08] D. L. Donoho, *Reflections on Compressed sensing*, IEEE Information theory society newsletter, desembre 2008.

- [Don89] D. L. Donoho i P. B. Stark *Uncertainty principles and signal recovery*, SIAM J. Applied Math., Vol. 49, pp. 906-931, 1989.
- [ElK04] N. El Karoui, *New results about Random Covariance Matrices and Statical Applications*, Stanford Ph. D Thesis, agost 2004.
- [Fra99] M. W. Frazier, *An introduction to Wavelets Through Linear Algebra*, Ed. Springer, 1999.
- [Gem80] S. Geman, *A limit theorem for the norm of random matrices*, Ann. Probab, Vol. 8, pp. 252-261, 1980.
- [Her96] E. Hernández i G. Weiss, *A first course on wavelets*, CRC Press LLC, 1996.
- [Hoe63] W. Hoeffding, *Probability Inequalities for Sums of Bounded Random Variables*, Journal of the America Statical Association, Vol. 58, Issue 301, pp 13-30, març 1963.
- [Kas77] B. Kashin, *The width of certain finite dimensional sets and classes of smooth functions*, Izvestia, Vol. 41, pp 334-351, 1977.
- [Led01] M. Ledux, *The concentration of measure phenomenoso*, Mathematical Surveys and Monographs, Vol. 89, American Mathematical Society, 1967.
- [Mar67] V. A. Marchenko i L. A. Pastur, *Distribution of eigenvalues in certain sets of random matrices*, Math. Sb. (N.S.), Vol. 72, pp. 407-535, 1967.
- [Rob67] G. Robinson i E.T. Whittaker, *Stirling's Approximation to the Factorial*, The Calculus of Observations: A Treatise on Numerical Mathematics, Dover, New York, pp 138-140, 1967.
- [Rud91] W. Rudin, *Functional Analysis*, 2a ed., McGraw-Hill, 1991.
- [She94] J.R. Shewchuk, *An introduction to the conjugate gradient method without the agonizing pain*, Manuscript, agost 1994.
- [Sil85] J. W. Silverstein, *The smallest eigenvalue of a large dimensional Wishart matrix*, Ann. Probab, Vol. 13, pp. 1364-1368, 1985.
- [Ste05] E.M. Stein i R. Shakarchi, *Real Analysis, Measure Theory, Integration & Hilbert Spaces*, Vol. 3, p.91, 1a ed., 2005.
- [Sza91] S. J. Szarek, *Conditions numbers of random matrices*, J. Complexity, Vol. 7, pp. 131-149, 1991.
- [Var84] S.R.S. Varadhan, *Large Deviations and Applications*, SIAM, 1984.