



Facultad
de
Ciencias

**Técnica de optimización para la
recuperación de imágenes**

(Optimization technique for image restoration)

Trabajo de Fin de Grado

para acceder al

GRADO EN MATEMÁTICAS

Autor: Carlos Laguillo García

Directora: Cecilia Pola Méndez

Santander - 22 de junio de 2015

*«Un matemático es un quijote moderno que lucha en un mundo real con
armas imaginarias».*

P. Corcho

Índice general

1. Introducción	5
2. Existencia y unicidad de solución	9
3. Condiciones de optimalidad	11
3.1. Unas nociones de cálculo subdiferencial	11
3.2. Condiciones de optimalidad para (PCG)	15
4. Un algoritmo de conjunto activo	19
4.1. Un algoritmo para (PCG)	20
4.2. Convergencia del algoritmo	24
4.3. Estudio de la degeneración	28
5. Recuperación de imágenes	33
5.1. Modelización del problema como (PCG)	33
5.2. Otra formulación equivalente	36
5.3. Ensayos numéricos	40

Resumen

En este trabajo vamos a estudiar un problema de programación cuadrática generalizada (con términos l_1 no diferenciables) que requiere algunos conocimientos de cálculo subdiferencial para determinar sus condiciones de optimalidad. Para su resolución numérica consideraremos un método de conjunto activo y probaremos su convergencia.

Por último aplicaremos lo aprendido a un problema de recuperación de imágenes donde los términos l_1 son útiles para la reducción del ruido en la imagen. Presentaremos algunos ensayos numéricos obtenidos con dos códigos, uno propio basado en el método de conjunto activo y quadprog (Optimization Toolbox, MATLAB) que implementa un algoritmo de puntos interiores.

Palabras clave: programación cuadrática generalizada, condiciones de optimalidad, subdiferencial, algoritmo de conjunto activo, recuperación de imágenes.

Abstract

In this work we are going to study a generalized quadratic programming problem (with non-differentiable l_1 terms) that requires some subdifferential calculus knowledge in order to determine its optimality conditions. To solve the problem we will consider an active-set method and its convergence properties.

Finally we will apply this study to an image restoration problem where l_1 terms are useful to reduce noise from image. We will present some numerical results obtained with two codes, our own implementation of the active-set algorithm and quadprog (Optimization Toolbox, MATLAB) that implements an interior-point algorithm.

Key words: Generalized quadratic programming, optimality conditions, subdifferential, active set, image restoration.

Capítulo 1

Introducción

La optimización puede definirse como la ciencia que busca una solución óptima a un problema formulado matemáticamente mediante una función que queremos minimizar teniendo las variables que cumplir un conjunto de igualdades y desigualdades. En general, un problema de optimización en dimensión finita puede escribirse como

$$(P) \begin{cases} \text{Minimizar} & f(x) \\ \text{sujeto a} & g_i(x) = 0, \quad 1 \leq i \leq m_i, \\ & g_i(x) \leq 0, \quad m_i < i \leq m_i + m_d, \end{cases}$$

con $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$. A la función f se le llama *función objetivo*. A las condiciones de igualdad y desigualdad de (P) se les llama *restricciones*, y al conjunto de puntos que las cumplen, *conjunto de puntos admisibles*.

Estos problemas se clasifican según las condiciones de regularidad que cumplan la función objetivo y las funciones de las restricciones (se habla así de programación lineal, programación cuadrática, optimización no diferenciable, ...) y sus aplicaciones abarcan un gran rango de posibilidades como la mejora de los procesos industriales o de algunos tratamientos médicos.

En el desarrollo de este trabajo estudiaremos un caso particular en el que la función objetivo es convexa y no diferenciable y las restricciones son lineales, junto con una aplicación, la recuperación de imágenes. Más concretamente

- En el capítulo 4 se encuentra una descripción detallada de un método de conjunto activo junto con los principales resultados teóricos del trabajo que nos garantizan la convergencia del algoritmo propuesto y un estudio del tratamiento de la degeneración.
- El capítulo 5 centra nuestra atención sobre el problema de recuperación de una imagen a partir de una distorsión de la misma producida por la alteración aleatoria de la intensidad en cada píxel de la imagen original. En este caso tenemos una simplificación de la formulación de (*PCG*), ya que podemos prescindir en la función objetivo de los términos asociados al último sumatorio, manteniendo los términos en los que aparece la función valor absoluto con el objetivo de amortiguar el ruido de la imagen. La función q vendrá dada por un término mínimo-cuadrático que mide la distancia a la imagen perturbada. También se encuentran en el capítulo 5 una serie de ensayos numéricos, a modo de ejemplo, obtenidos con dos métodos distintos, el propuesto en el capítulo 4 y uno de puntos interiores implementado en *quadprog* (MATLAB).

Capítulo 2

Existencia y unicidad de solución

La primera pregunta que debemos plantearnos en el estudio de nuestro problema (*PCG*) (y en general de cualquier problema matemático) es si existe alguna solución, y en caso de existir, si dicha solución es única. Para ello vamos a introducir el siguiente concepto.

Definición Dada una función $f : K \subset \mathbb{R}^n \rightarrow \mathbb{R}$ decimos que f es *coerciva* sobre K si para toda sucesión $(x_m)_{m \in \mathbb{N}} \subset K$ con $\|x_m\| \xrightarrow{m \rightarrow +\infty} +\infty$ se tiene que $f(x_m) \xrightarrow{m \rightarrow +\infty} +\infty$.

Tenemos los siguientes resultados:

Teorema 2.0.1. *Dado un problema de optimización*

$$(P) \begin{cases} \text{Minimizar} & f(x) \\ \text{sujeto a} & x \in K \end{cases} ,$$

donde $f : K \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ es una función continua y coerciva sobre K , siendo $K \neq \emptyset$ un cerrado, entonces existe al menos una solución global para (P) .

Teorema 2.0.2. *Dado un problema (P) como en el enunciado anterior, si f es estrictamente convexa sobre K , K es convexo y existe solución para (P) , entonces dicha solución es única.*

Puede encontrarse una prueba de ambos resultados en [1].

En el caso que nos ocupa, (*PCG*), K es siempre cerrado y la función

a minimizar, F , continua, luego basta ver que F es coerciva para tener garantizada la existencia de alguna solución.

Para ello observemos que por ser H simétrica se tiene que

$$x^T H x \geq \gamma_{min} \|x\|_2^2, \quad \forall x \in \mathbb{R}^n, \quad (2.1)$$

donde γ_{min} es el menor valor propio de la matriz H . Además H es definida positiva, luego $\gamma_{min} > 0$.

Por otro lado, de la desigualdad de Cauchy-Schwartz tenemos también que

$$p^T x \geq -\|p\|_2 \|x\|_2, \quad \forall x \in \mathbb{R}^n. \quad (2.2)$$

Ahora, usando (2.1), (2.2) y que $h_j(x) \geq 0$ para todo $x \in \mathbb{R}^n$, obtenemos la siguiente desigualdad:

$$F(x) \geq \alpha \|x\|_2 \left[\frac{\gamma_{min}}{2} \|x\|_2 - \|p\|_2 \right].$$

Por tanto F es coerciva.

En cuanto a la unicidad de solución, basta observar que F es estrictamente convexa por ser suma de funciones convexas y la matriz H definida positiva (de donde q es una función estrictamente convexa).

Capítulo 3

Condiciones de optimalidad

En este capítulo vamos a desarrollar las bases teóricas que sustentan la parada satisfactoria de nuestro algoritmo, con el objetivo puesto en establecer unos criterios (*condiciones de optimalidad*) viables a la hora de programar y que sirven para decidir si un punto es solución de (*PCG*). Estos criterios cuando las funciones del problema a tratar son derivables involucran las derivadas de las mismas. Como ya hemos mencionado, nuestro funcional contiene términos no diferenciables, por lo tanto no podemos aplicar muchos de los resultados vistos en cursos anteriores, por ello vamos a dedicar una sección previa al cálculo subdiferencial de funciones convexas.

3.1. Unas nociones de cálculo subdiferencial

Consideremos la siguiente definición de función convexa:

Definición Una función $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$, no idénticamente $+\infty$, se dice que es convexa si

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \quad \forall t \in (0, 1), \quad \forall x, y \in \mathbb{R}^n.$$

Usaremos la notación habitual para el *dominio* efectivo de f :

$$\text{dom}(f) = \{x \in \mathbb{R}^n : f(x) < +\infty\}.$$

Notemos que toda función convexa definida sobre un conjunto convexo,

$f : K \rightarrow \mathbb{R}$, puede extenderse a \mathbb{R}^n considerando $\bar{f} : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ dada por

$$\bar{f}(x) = \begin{cases} f(x) & \text{si } x \in K, \\ +\infty & \text{en otro caso.} \end{cases}$$

Definición Se dice que una función $f : \mathbb{R}^n \rightarrow (-\infty, \infty]$ convexa es *subdiferenciable* en $x \in \mathbb{R}^n$, si existe $x^* \in \mathbb{R}^n$ tal que:

$$\langle x^*, u - x \rangle + f(x) \leq f(u), \quad \forall u \in \mathbb{R}^n.$$

A x^* se le llama *subgradiente* de f en x y al conjunto de todos los subgradietes de f en x se le dice *subdiferencial* de f en x y se denota $\partial f(x)$.

Notemos que $\partial f(x) = \emptyset$ si $f(x) = +\infty$.

Uno de los principales impulsores del desarrollo del cálculo subdiferencial y sus aplicaciones fue Rockafellar ([8]).

El resultado siguiente nos muestra cómo la noción de subdiferencial es una generalización, para funciones convexas, del concepto de diferencial.

Proposición 3.1.1. *Sea $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ una función convexa y diferenciable en $x \in \text{dom}(f)$. Entonces*

$$\partial f(x) = \{\nabla f(x)\}.$$

Demostración En primer lugar, probemos que $\nabla f(x) \in \partial f(x)$. Sea $u \in \mathbb{R}^n$, como f es diferenciable en x :

$$\langle \nabla f(x), u - x \rangle = \lim_{t \downarrow 0} \frac{f(x + t(u - x)) - f(x)}{t}.$$

Y por ser f convexa:

$$f(x + t(u - x)) \leq tf(u) + (1 - t)f(x), \quad \forall t \in (0, 1).$$

De donde se concluye:

$$\lim_{t \downarrow 0} \frac{f(x + t(u - x)) - f(x)}{t} \leq f(u) - f(x).$$

Con lo cual obtenemos nuestro objetivo:

$$\langle \nabla f(x), u - x \rangle + f(x) \leq f(u), \quad \forall u \in \mathbb{R}^n. \quad (3.1)$$

Supongamos ahora que $x^* \in \partial f(x)$ y veamos que $x^* = \nabla f(x)$. Usando la definición de subgradiente se tiene

$$\langle x^*, u - x \rangle + f(x) \leq f(u), \quad \forall u \in \mathbb{R}^n.$$

En particular, dado $w \in \mathbb{R}^n$ podemos tomar $u = x + \lambda w$ con $\lambda > 0$ y tendremos que

$$\langle x^*, w \rangle \leq \frac{f(x + \lambda w) - f(x)}{\lambda}, \quad \forall \lambda > 0. \quad (3.2)$$

Si pasamos al límite en la expresión anterior con $\lambda \rightarrow 0^+$,

$$\langle x^*, w \rangle \leq \langle \nabla f(x), w \rangle.$$

Y así

$$\langle \nabla f(x) - x^*, w \rangle \geq 0, \quad \forall w \in \mathbb{R}^n.$$

Y por tanto se concluye que $x^* = \nabla f(x)$. \square

Veamos algunos ejemplos de subdiferenciales para funciones que aparecen en la formulación de nuestro problema (*PCG*) o que serán de interés más adelante.

- La función $f : \mathbb{R} \rightarrow \mathbb{R}$ dada por $f(x) = \max\{0, x\}$.
 - I) Caso $x > 0$. En $(0, +\infty)$, $f(x) = x$. Al ser diferenciable y convexa, aplicando la proposición anterior, $\partial f(x) = \{f'(x)\} = \{1\}$.
 - II) Caso $x < 0$. En $(-\infty, 0)$, $f(x) = 0$. Usando los mismos argumentos que en el caso anterior, $\partial f(x) = \{f'(x)\} = \{0\}$.
 - III) Caso $x = 0$. Se tienen las siguientes equivalencias:

$$\begin{aligned} x^* \in \partial f(0) &\Leftrightarrow x^* u \leq \max\{0, u\}, \quad \forall u \in \mathbb{R} \Leftrightarrow \left\{ \begin{array}{l} x^* u \leq u \quad \text{si } u > 0 \\ x^* u \leq 0 \quad \text{si } u \leq 0 \end{array} \right\} \\ &\Leftrightarrow x^* \in [0, 1] \end{aligned}$$

- La función $f : \mathbb{R} \rightarrow \mathbb{R}$ dada por $f(x) = |x|$.
 - I) Caso $x > 0$. En $(0, +\infty)$, $f(x) = x$. Al ser diferenciable y convexa, aplicando la proposición anterior, $\partial f(x) = \{f'(x)\} = \{1\}$.

- II) Caso $x < 0$. En $(-\infty, 0)$, $f(x) = -x$. Razonando del mismo modo que antes, $\partial f(x) = \{f'(x)\} = \{-1\}$.
- III) Caso $x = 0$. Se tiene:

$$\begin{aligned} x^* \in \partial f(0) &\Leftrightarrow x^*u \leq |u| \Leftrightarrow \begin{cases} x^*u \leq u & \text{si } u > 0 \\ x^*u \leq -u & \text{si } u \leq 0 \end{cases} \\ &\Leftrightarrow x^* \in [-1, 1]. \end{aligned}$$

- Dado $B \subset \mathbb{R}^n$, $B \neq \emptyset$, sea la función $\chi_B : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ dada por

$$\chi_B(x) = \begin{cases} 0 & \text{si } x \in B, \\ +\infty & \text{si } x \notin B, \end{cases}$$

a la que llamaremos *función indicador de B*. Veamos cómo es la subdiferencial de esta función en dos casos particulares que encontraremos más adelante.

$B = \{0\}$ En este caso tenemos las siguientes posibilidades:

- I) Si $x = 0$ se tiene que: $x^* \in \partial\chi_{\{0\}}(0) \Leftrightarrow \langle x^*, 0 \rangle \leq 0$. Luego $\partial\chi_{\{0\}}(0) = \mathbb{R}^n$.
- II) Si $x \neq 0$, $\partial\chi_{\{0\}}(x) = \emptyset$.

$B = (-\infty, 0]$ En este caso tenemos:

- I) Si $x < 0$ entonces: $x^* \in \partial\chi_{\{B\}}(x) \Leftrightarrow x^*(u - x) \leq 0, \forall u \in B$. Luego $\partial\chi_{\{B\}}(0) = \{0\}$.
- II) Si $x = 0$ se tiene que: $x^* \in \partial\chi_{\{B\}}(0) \Leftrightarrow x^*u \leq 0, \forall u \in B$. Luego $\partial\chi_{\{B\}}(0) = [0, +\infty)$.
- III) Si $x > 0$, $\partial\chi_{\{B\}}(x) = \emptyset$.

En general no podemos decir que la subdiferencial de la suma de funciones sea la suma de las respectivas subdiferenciales, sin embargo para las funciones de (PCG) esto sí que es cierto. También tenemos una regla de la cadena para la subdiferencial, más restrictiva que la usual, en el sentido en que impone más condiciones de regularidad sobre las funciones involucradas, pero que será válida para las funciones que nos ocupan (ver [7]).

3.2. Condiciones de optimalidad para (PCG)

A continuación establecemos una primera condición de optimalidad para un problema de optimización sin restricciones.

Proposición 3.2.1. *Sea $\bar{x} \in \mathbb{R}^n$. Sean $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función derivable en \bar{x} y $\varphi : \mathbb{R}^n \rightarrow (-\infty, \infty]$ una función convexa. Dado el problema:*

$$\underset{x \in \mathbb{R}^n}{\text{Minimizar}} (f(x) + \varphi(x)),$$

si \bar{x} es una solución, entonces

$$0 \in \nabla f(\bar{x}) + \partial\varphi(\bar{x}). \quad (3.3)$$

Demostración Por ser \bar{x} solución del problema, dado $s \in \mathbb{R}^n$ cualquiera, para t suficientemente pequeño tenemos que:

$$f(\bar{x} + ts) + \varphi(\bar{x} + ts) \geq f(\bar{x}) + \varphi(\bar{x}).$$

Ahora, si $t > 0$

$$\frac{f(\bar{x} + ts) - f(\bar{x})}{t} + \frac{\varphi(\bar{x} + ts) - \varphi(\bar{x})}{t} \geq 0. \quad (3.4)$$

Además, si $t < 1$, por ser φ convexa tenemos que

$$\bar{x} + ts = t(\bar{x} + s) + (1 - t)\bar{x} \Rightarrow \varphi(\bar{x} + ts) \leq t\varphi(\bar{x} + s) + (1 - t)\varphi(\bar{x}).$$

Por tanto,

$$\frac{\varphi(\bar{x} + ts) - \varphi(\bar{x})}{t} \leq \frac{t\varphi(\bar{x} + s) + (1 - t)\varphi(\bar{x}) - \varphi(\bar{x})}{t} = \varphi(\bar{x} + s) - \varphi(\bar{x}).$$

De donde, volviendo a (3.4) se deduce que:

$$\frac{f(\bar{x} + ts) - f(\bar{x})}{t} + \varphi(\bar{x} + s) - \varphi(\bar{x}) \geq 0.$$

Y tomando el límite con t decreciendo hacia 0.

$$\lim_{t \rightarrow 0} \frac{f(\bar{x} + ts) - f(\bar{x})}{t} + \varphi(\bar{x} + s) - \varphi(\bar{x}) \geq 0.$$

Aplicando ahora la definición de derivada direccional se tiene:

$$\langle \nabla f(\bar{x}), s \rangle + \varphi(\bar{x} + s) - \varphi(\bar{x}) \geq 0, \quad \forall s \in \mathbb{R}^n.$$

Y, por tanto:

$$\varphi(v) \geq -\langle \nabla f(\bar{x}), v - \bar{x} \rangle + \varphi(\bar{x}), \quad \forall v \in \mathbb{R}^n.$$

Luego $-\nabla f(\bar{x}) \in \partial\varphi(\bar{x})$ y tenemos (3.3). \square

Con estos resultados podemos demostrar el principal teorema de esta sección, las condiciones de optimalidad necesarias y suficientes de nuestro problema (PCG).

Teorema 3.2.2. *Sea \bar{x} un punto admisible para (PCG). Entonces \bar{x} es la solución de (PCG) si, y sólo si*

$$\exists \bar{\mu} \in \mathbb{R}^{m_{if}+m_{df}} \text{ y } \exists \bar{\lambda} \in \mathbb{R}^{m_i+m_d} \text{ tales que:}$$

$$\alpha \nabla q(\bar{x}) + \bar{C}\bar{\mu} + C\bar{\lambda} = 0, \quad (3.5)$$

$$\bar{\mu}_j \in \begin{cases} \{1\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j > 0, \\ [-1, 1] & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j = 0, \quad 1 \leq j \leq m_{if}, \\ \{-1\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j < 0, \end{cases} \quad (3.6)$$

$$\bar{\mu}_j \in \begin{cases} \{1\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j > 0, \\ [0, 1] & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j = 0, \quad m_{if} < j \leq m_{if} + m_{df}, \\ \{0\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j < 0, \end{cases} \quad (3.7)$$

$$\bar{\lambda}_j (c_j^T \bar{x} - b_j) = 0, \quad m_i < j \leq m_i + m_d, \quad (3.8)$$

$$\bar{\lambda}_j \geq 0, \quad m_i < j \leq m_i + m_d. \quad (3.9)$$

Demostración En primer lugar veamos que si \bar{x} es solución de (PCG) también lo es de un problema de los considerados en la Proposición 3.2.1 Para ello, consideramos la función $G : \mathbb{R}^n \rightarrow \mathbb{R}^{m_i}$ definida por

$$G(x) = C_{m_i}^T x - B_{m_i},$$

donde C_{m_i} es la matriz cuyas columnas contienen los coeficientes de las restricciones de igualdad y B_{m_i} el vector con los términos independientes asociados a dichas restricciones, las funciones $\Psi_j(x) = c_j^T x - b_j$ y el funcional

$$g(x) = \alpha q(x) + \varphi(x)$$

con

$$\varphi(x) = h(x) + (\chi_0 \circ G)(x) + \sum_{j=m_i+1}^{m_i+m_d} (\chi_{(-\infty,0]} \circ \Psi_j)(x).$$

Entonces, si \bar{x} es solución de (PCG), también lo es del problema sin restricciones

$$(P_1) \quad \text{Min}_{x \in \mathbb{R}^n} g(x).$$

Como q es derivable en \bar{x} y φ es convexa, aplicando la proposición 3.2.1 se tiene

$$0 \in \alpha \nabla q(\bar{x}) + \partial \varphi(\bar{x}).$$

Aplicando la aditividad de la subdiferencial se tiene

$$0 \in \alpha \nabla q(\bar{x}) + \partial h(\bar{x}) + \partial(\chi_0 \circ G)(\bar{x}) + \partial\left(\sum_{j=m_i+1}^{m_i+m_d} \chi_{(-\infty,0]} \circ \Psi_j\right)(\bar{x}).$$

Ahora aplicando la regla de la cadena y el Lema de Farkas (ver [7]) tenemos

$$0 \in \alpha \nabla q(\bar{x}) + \partial h(\bar{x}) + C_{m_i} \partial \chi_0(0) + \sum_{j=m_i+1}^{m_i+m_d} (\partial \chi_{(-\infty,0]}(c_j^T \bar{x} - b_j)) c_j. \quad (3.10)$$

Para calcular $\partial h(\bar{x})$, volvemos a aplicar la aditividad de la subdiferencial y la regla de la cadena. Si tomamos $\phi(t) = |t|$ y $\gamma_j(t) = \bar{c}_j^T x - \bar{b}_j$ para $j = 1, \dots, m_{if}$, tenemos que $h_j = \phi \circ \gamma_j$ y obtenemos

$$\partial h_j(\bar{x}) = \partial(\phi \circ \gamma_j)(\bar{x}) = \bar{c}_j \partial \phi(\bar{c}_j^T \bar{x} - \bar{b}_j).$$

Y, como vimos en los ejemplos de cálculo de subdiferenciales,

$$\partial \phi(\bar{c}_j^T \bar{x} - \bar{b}_j) = \begin{cases} \{1\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j > 0, \\ [-1, 1] & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j = 0, \\ \{-1\} & \text{si } \bar{c}_j^T \bar{x} - \bar{b}_j < 0. \end{cases}$$

Obteniendo así (3.6) tomando $\bar{\mu}_i \in \partial\phi(\bar{c}_j^T \bar{x} - \bar{b}_j)$.

De forma análoga podemos calcular $\partial h_j(\bar{x})$ para $j > m_{if}$ y obtener (3.7).

Volviendo a (3.10), tomando $\bar{\lambda}^{(1)} \in \partial\chi_0(0) = \mathbb{R}^{m_i}$, $\bar{\lambda}^{(2)} = (\lambda_{m_i+1}, \dots, \lambda_{m_i+m_d})$

con $\lambda_i \in \partial\chi_{(-\infty, 0]}(c_i^T \bar{x} - b_i)$, y $\bar{\lambda} = (\bar{\lambda}^{(1)T}, \bar{\lambda}^{(2)T})^T$, se tienen (3.5), (3.8) y (3.9).

Con lo cual queda probada una de las implicaciones.

Para el recíproco, si tenemos $0 = \alpha \nabla q(\bar{x}) + \bar{C}\bar{\mu} + C\bar{\lambda} \in \partial g(\bar{x})$, aplicando la definición de subdiferencial tenemos que $g(\bar{x}) \leq g(u)$, $\forall u \in \mathbb{R}^n$. De donde, usando que $g(x) = F(x)$ para cualquier punto admisible de (PCG) y que, por hipótesis, \bar{x} es uno de esos puntos admisibles, tenemos que \bar{x} es solución de (PCG) .

Capítulo 4

Un algoritmo de conjunto activo

En este capítulo consideraremos un algoritmo de conjunto activo para resolver el problema de optimización (*PCG*). En ([7]) se tiene un esbozo del mismo. Nosotros estudiaremos en detalle sus propiedades teóricas y haremos una implementación en MATLAB adaptada al caso especial del problema de recuperación de imágenes que consideraremos más adelante.

El algoritmo que vamos a implementar genera una sucesión de puntos admisibles, $(x^k)_{k \in \mathbb{N}} \subseteq K$, hasta que se obtiene un punto cumpliendo las condiciones de optimalidad del teorema 3.2.2. Cada x^k llevará asociado un *conjunto de trabajo*, J_k , que es un conjunto de índices de restricciones que se encuentran activas en ese punto, es decir se cumplen con igualdad. El conjunto J_k se descompondrá en dos subconjuntos, $J_k = I_k \cup \bar{I}_k$, para distinguir los índices asociados a restricciones en K , I_k , y los índices asociados a restricciones relajadas del funcional, \bar{I}_k . La matriz asociada al conjunto de trabajo, $A_k \in \mathbb{R}^{n \times m_k}$, será la matriz que contiene en sus columnas los coeficientes de las restricciones del conjunto de trabajo y debe ser de rango máximo. Para cada matriz A_k dispondremos de su factorización QR , de forma que se cumpla:

$$\begin{aligned} A_k &= Q_k R_k, \\ Q_k &= [Y_k \ Z_k], \\ R_k &= \begin{bmatrix} \hat{R}_k \\ 0 \end{bmatrix}, \end{aligned}$$

donde $Q_k \in \mathbb{R}^{n \times n}$ es una matriz ortogonal, Y_k es una matriz cuyas columnas forman una base del subespacio del rango de A_k y las columnas de Z_k forman una base del subespacio del núcleo de A_k^T y $\hat{R}_k \in \mathbb{R}^{m_k \times m_k}$ es una matriz triangular superior de rango máximo.

Llamaremos *Hessiano reducido*, y lo denotaremos H_{Z_k} , a la matriz $Z_k^T H Z_k$, donde H es la matriz asociada a la parte cuadrática del funcional a minimizar en (PCG).

Denotaremos por V_k al conjunto de índices asociados a restricciones del funcional que no están en el conjunto de trabajo.

4.1. Un algoritmo para (PCG)

Propondremos un algoritmo de conjunto activo que es una generalización del método visto en la asignatura Optimización I para programación cuadrática convexa estándar (sin términos de penalización).

PASO 0:

Si el usuario no suministra un punto admisible, calcular uno, x^0 . Poner $k = 0$.

Construir el primer conjunto de trabajo, I_0 , con las restricciones activas en x^0 que sean linealmente independientes y calcular la correspondiente factorización QR .

PASO 1:

Definimos el subgradiente como $g^k = \alpha(Hx^k + p) + \sum_{j \in V_k} \eta_j^k \bar{c}_j$, donde

$$\eta_j^k = \begin{cases} 1 & \text{si } \bar{c}_j^T x^k > \bar{b}_j, \\ -1 & \text{si } \bar{c}_j^T x^k < \bar{b}_j \\ 0 & \text{en otro caso.} \end{cases} \quad \text{y } 1 \leq j \leq m_{if}, \quad (4.1)$$

Si $Z_k^T g^k \neq 0$, seguir en el paso 2. En caso contrario calcular los candidatos a multiplicadores de Lagrange, $\bar{\lambda}^{(k)}$, que verifican el sistema de ecuaciones lineales siguiente:

$$A_k \bar{\lambda}^{(k)} = -g^k. \quad (4.2)$$

Sea $\bar{\lambda}^{(k)} \in \mathbb{R}^{m_i + m_d + \bar{m}_k}$ reconstruido a partir de la solución del sistema anterior, tomando $\bar{\lambda}_i^k = 0$ para las restricciones de K que no están en el

conjunto de trabajo y siendo \bar{m}_k el número de restricciones del funcional que están en el conjunto de trabajo.

- Si $\bar{\lambda}^{(k)}$ cumple las condiciones del teorema 3.2.2 entonces x^k es la solución de (PCG) y el algoritmo finaliza satisfactoriamente.
- En otro caso, eliminar del conjunto de trabajo la restricción asociada a un multiplicador de Lagrange que viola la correspondiente condición de optimalidad (3.6), (3.7) o (3.9) eligiendo o bien el primero que lo hace (*estrategia 2*) o bien el mayor incumplidor (*estrategia 1*). Para esta última tomar las siguientes expresiones que varían según sea la restricción a considerar:

$$\gamma_i^k = \begin{cases} \bar{\lambda}_i^k & \text{si es de desigualdad y de } K, \\ 1 - |\bar{\lambda}_i^k| & \text{si es de igualdad asociada a } F, \\ \min\{\bar{\lambda}_i^k, 1 - \bar{\lambda}_i^k\} & \text{si es de desigualdad asociada a } F. \end{cases}$$

Sea S^λ el menor de todos estos valores.

Si la restricción eliminada está asociada a un término del funcional F , modificar también el gradiente de la forma

$$\bar{g}^k = g^k + \sigma^k \bar{c}_{j_0}, \quad (4.3)$$

donde σ^k viene dado por

$$\sigma^k = \begin{cases} 1 & \text{si } \bar{\lambda}_{j_0}^k > 1, \\ 0 & \text{si } \bar{\lambda}_{j_0}^k < 0 \quad \text{y la restricción es de desigualdad,} \\ -1 & \text{si } \bar{\lambda}_{j_0}^k < -1 \quad \text{y la restricción es de igualdad,} \end{cases}$$

siendo $\bar{\lambda}_{j_0}^k$ el multiplicador asociado a S^λ .

PASO 2: Calcular una dirección como $d^k = Z_k d_{Z_k}$ siendo d_{Z_k} la solución del sistema

$$H_{Z_k} d_{Z_k} = -\frac{1}{\alpha} (Z_k)^T \bar{g}^k, \quad (4.4)$$

donde $\bar{g}_k = g_k$ en caso de no haberse aplicado (4.3).

PASO 3: Estudiar la degeneración y calcular el paso, ρ_k , con información

de las restricciones que no están en el conjunto de trabajo mediante la siguiente fórmula

$$\rho_k = \min\{1, \rho_l, \rho_{lf}\}, \quad (4.5)$$

donde ρ_l está asociado a restricciones de K :

$$\rho_l = \min_{j \notin I_k, c_j^T d^k > 0} \left\{ \frac{b_j - c_j^T x^k}{c_j^T d^k} \right\}$$

y ρ_{lf} corresponde a restricciones asociadas al funcional F :

$$\rho_{lf} = \min_{j \in V_{k,1} \cup V_{k,2} \cup V_{k,3}} \rho_j, \quad (4.6)$$

siendo

$$\begin{aligned} V_{k,1} &= \{i \in V_k : \bar{c}_i^T x^k \leq \bar{b}_i, \bar{c}_i^T d^k > 0\}, \\ V_{k,2} &= \{i \in V_k : \bar{c}_i^T x^k > \bar{b}_i, \bar{c}_i^T d^k < 0\}, \\ V_{k,3} &= \{i \in V_k : \bar{c}_i^T x^k = \bar{b}_i, \bar{c}_i^T d^k < 0, i \leq m_{if}\}, \\ \rho_j &= \frac{\bar{b}_j - \bar{c}_j^T x^k}{\bar{c}_j^T d^k}. \end{aligned}$$

Si $\rho_k < 1$, buscar un índice en el que se alcance el mínimo y añadir la correspondiente restricción al conjunto de trabajo. Tomar $x^{k+1} = x^k + \rho_k d^k$ y volver al paso 1.

Observaciones

- 1) Nuestro algoritmo requiere de un punto inicial, $x^0 \in K$, para comenzar. Este punto puede bien ser suministrado por el usuario, o bien puede ser calculado de diversas formas, por ejemplo resolviendo un problema de optimización no diferenciable como el siguiente:

$$(P_0) \left\{ \begin{array}{l} \text{Minimizar} \quad f(x) = \sum_{j=m_i+1}^{m_i+m_d} \max\{0, c_j^T x - b_j\} \\ \text{sujeto a} \quad c_i^T x = b_j \quad 1 \leq i \leq m_i. \end{array} \right.$$

La resolución numérica de (P_0) (ver [7]) nos proporcionará, junto con un punto inicial admisible, un conjunto de trabajo y la correspondiente factorización QR de su matriz asociada, lo cual nos permite ahorrar cálculos en nuestro algoritmo.

- II) En la práctica, la nulidad de $Z_k^T g^k$ en el **PASO 1** se comprueba mediante la comparación de su norma euclídea con una pequeña constante dependiente de la precisión de la máquina a la que llamaremos *tolerancia*.
- III) En la práctica, para calcular los candidatos a multiplicadores de Lagrange que aparecen en (4.2) usaremos la factorización QR de A_k y resolveremos el sistema triangular

$$\hat{R}_k \lambda = -Y_k^T g^k.$$

En cuanto al sistema (4.4), utilizaremos la factorización de Cholesky del Hessiano reducido, $H_{Z_k} = LL^T$, para transformar el sistema original en dos sistemas de ecuaciones lineales cuyas matrices asociadas son triangulares:

$$Ly = -\frac{1}{\alpha}(Z_k)^T \bar{g}^k,$$

$$L^T d_{Z_k} = y.$$

- IV) La modificación del subgradiente en la ecuación (4.3) del **PASO 1** ha de ser coherente con el signo de $\bar{c}_{j_0}^T d^k$.

Notemos que $a_i^T d^k = 0$ para $i \in J_k \setminus \{j_0\}$, siendo a_i la i -ésima columna de A_k , y por tanto multiplicando la ecuación (4.2) por d^k tenemos que

$$(g^k)^T d^k = -\bar{\lambda}_{j_0}^k \bar{c}_{j_0}^T d^k,$$

luego

$$(g^k)^T d^k + \sigma^k \bar{c}_{j_0}^T d^k - \sigma^k \bar{c}_{j_0}^T d^k = -\bar{\lambda}_{j_0}^k \bar{c}_{j_0}^T d^k,$$

de donde, agrupando términos y utilizando (4.3),

$$(\bar{g}^k)^T d^k = (\sigma^k - \bar{\lambda}_{j_0}^k) \bar{c}_{j_0}^T d^k,$$

obteniendo así

$$\bar{c}_{j_0}^T d^k = \frac{(\bar{g}^k)^T d^k}{(\sigma^k - \bar{\lambda}_{j_0}^k)}.$$

Ahora, para determinar el signo de la expresión anterior, notemos que el numerador es negativo ya que $(\bar{g}^k)^T d^k = -\alpha(d^k)^T H d^k < 0$. Teniendo en cuenta el denominador distinguimos tres casos:

- Si $\bar{\lambda}_{j_0}^k > 1$, entonces $\sigma^k = 1$, y por lo tanto $\bar{c}_{j_0}^T d^k > 0$.
- Si $\bar{\lambda}_{j_0}^k < 0$ y $j_0 > m_{if}$, entonces $\sigma^k = 0$, luego $\bar{c}_{j_0}^T d^k < 0$.
- Si $\bar{\lambda}_{j_0}^k < -1$ y $j_0 \leq m_{if}$, $\sigma^k = -1$, de donde $\bar{c}_{j_0}^T d^k < 0$.

v) Siempre que se añaden o eliminan restricciones del conjunto de trabajo, para ahorrar cálculos, conviene modificar la factorización QR de la matriz asociada al conjunto de trabajo y la factorización de Cholesky del Hessiano reducido en lugar de calcular completamente estas factorizaciones.

vi) En el **PASO 3**, al calcular el desplazamiento no se considera la restricción eliminada en la misma iteración si éste fuera el caso.

vii) Notemos que el algoritmo siempre genera puntos admisibles. Para ello veamos que $x^{k+1} = x^k + \rho_k d^k \in K$.

Si $i \in I_k$, entonces por ser d^k combinación lineal de una base del subespacio del núcleo de A_k^T se tiene que $c_i^T d^k = 0$ y por tanto

$$c_i^T x^{k+1} = c_i^T x^k + \rho_k c_i^T d^k = b_i.$$

Si $i \notin I_k$, teniendo en cuenta el cálculo de ρ_k en el **PASO 3**, es claro que $\rho_k c_i^T d^k \leq b_i - c_i^T x^k$, de donde se deduce que

$$c_i^T x^{k+1} = c_i^T x^k + \rho_k c_i^T d^k \leq b_i.$$

4.2. Convergencia del algoritmo

En esta sección probaremos un par de resultados antes de obtener la convergencia del algoritmo propuesto. Empezamos por establecer que el desplazamiento en la dirección calculada nos conduce a decrecimiento de la función objetivo de (PCG) .

Proposición 4.2.1. *Si el desplazamiento, ρ_k , no es nulo, la dirección d^k calculada en el **PASO 2** es una dirección de descenso que verifica:*

$$F(x^k + \rho d^k) < F(x^k) \text{ para todo } \rho \in (0, \rho_k].$$

Demostración Para comparar $F(x^k + \rho d^k)$ y $F(x^k)$ (recordemos que $F(x) = \alpha q(x) + h(x)$) determinemos los sumandos de $h(x^k + \rho d^k)$ que sean no nulos. Notemos que podemos prescindir de los sumandos asociados a restricciones que son activas en x^k ($\bar{c}_j^T x^k = \bar{b}_j$) excepto para la eliminada durante la iteración, bien porque están en el conjunto de trabajo ($\bar{c}_j^T d^k = 0$) o bien porque $\rho_k > 0$. Para el resto de los sumandos, comenzamos con aquellos asociados a restricciones de igualdad y, usando $\Phi_1^j(\rho) = |\bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j|$ y $\epsilon_j = \frac{\bar{b}_j - \bar{c}_j^T x^k}{\bar{c}_j^T d^k} > 0$, observamos los siguientes casos:

I) Si $\bar{c}_j^T x^k > \bar{b}_j$, se tiene que:

- Si $\bar{c}_j^T d^k \geq 0$, $\Phi_1^j(\rho) = \bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j$, $\forall \rho \in (0, +\infty)$.
- Si $\bar{c}_j^T d^k < 0$, $\Phi_1^j(\rho) = \bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j$, $\forall \rho \in (0, \epsilon_j]$.

II) Si $\bar{c}_j^T x^k < \bar{b}_j$, entonces:

- Si $\bar{c}_j^T d^k \leq 0$, $\Phi_1^j(\rho) = -(\bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j)$, $\forall \rho \in (0, +\infty)$.
- Si $\bar{c}_j^T d^k > 0$, $\Phi_1^j(\rho) = -(\bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j)$, $\forall \rho \in (0, \epsilon_j]$.

Consideramos ahora el resto de los términos asociados a restricciones de desigualdad. Con la notación $\Phi_2^j(\rho) = \max(0, \bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j)$ y definiendo ϵ_j del mismo modo que antes, observamos los siguientes casos:

I) Si $\bar{c}_j^T x^k > \bar{b}_j$, tenemos:

- Si $\bar{c}_j^T d^k \geq 0$, $\Phi_2^j(\rho) = \bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j$, $\forall \rho \in (0, +\infty)$.
- Si $\bar{c}_j^T d^k < 0$, $\Phi_2^j(\rho) = \bar{c}_j^T(x^k + \rho d^k) - \bar{b}_j$, $\forall \rho \in (0, \epsilon_j]$.

II) Si $\bar{c}_j^T x^k < \bar{b}_j$

- Si $\bar{c}_j^T d^k \leq 0$, $\Phi_2^j(\rho) = 0$, $\forall \rho \in (0, +\infty)$.
- Si $\bar{c}_j^T d^k > 0$, $\Phi_2^j(\rho) = 0$, $\forall \rho \in (0, \epsilon_j]$.

De este modo, usando (4.1) y (4.6), tenemos que si $0 < \rho \leq \rho_{lf}$ entonces

$$\begin{aligned}
F(x^k + \rho d^k) &= \alpha q(x^k + \rho d^k) + \sum_{j=1}^{m_{if}} \Phi_1^j(\rho) + \sum_{j=m_{if}+1}^{m_{if}+m_{df}} \Phi_2^j(\rho) = \\
&= F(x^k) + \rho \left[\alpha \nabla q(x^k) + \sum_{j \in V_k} \eta_j^k \bar{c}_j \right]^T d^k + \frac{\alpha \rho^2}{2} (d^k)^T H d^k = \\
&= F(x^k) + \rho (\bar{g}^k)^T d^k + \frac{\alpha \rho^2}{2} (d^k)^T H d^k. \tag{4.7}
\end{aligned}$$

Notemos que si en la iteración se ha utilizado (4.3), las igualdades anteriores se verifican con $\eta_{j_0}^k = \sigma^k$.

Ahora, usando (4.4) se deduce $(d^k)^T \bar{g}^k = -\alpha (d^k)^T H d^k$ y por tanto (4.7) puede reescribirse como

$$F(x^k + \rho d^k) = F(x^k) + \alpha \rho \left(-1 + \frac{\rho}{2} \right) (d^k)^T H d^k. \tag{4.8}$$

Así, usando que H es definida positiva y que d^k es no nulo, se concluye que $F(x^k + \rho d^k) < F(x^k)$, $\forall \rho \in (0, \min \{2, \rho_{lf}\}]$. \square

En el siguiente resultado analizamos el interés del paso unidad en (4.5).

Proposición 4.2.2. *Si $\rho_k = 1$, x^{k+1} es la solución del problema*

$$(P_{aux}) \left\{ \begin{array}{l} \text{Minimizar } F(x) = \alpha q(x) + \sum_{j \in V_k} h_j(x) \\ \text{sujeto a } \begin{array}{ll} c_j^T x = b_j & \text{si } j \in I_k, \\ \bar{c}_j^T x = \bar{b}_j & \text{si } j \in \bar{I}_k, \\ c_j^T x \leq b_j & \text{si } j \notin I_k. \end{array} \end{array} \right.$$

Demostración Veamos si se cumplen las condiciones suficientes de optimalidad. Aplicando (4.4) obtenemos:

$$Z_k^T \left[\alpha (H Z_k d_{Z_k} + H x^k + p) + \sum_{i \in V_k} \eta_i^k \bar{c}_i \right] = 0.$$

De donde se deduce que existen $\lambda_j \in \mathbb{R}$ para $j \in J_k = I_k \cup \bar{I}_k$ cumpliendo

$$\alpha [H(x^k + d^k) + p] + \sum_{i \in V_k} \eta_i^k \bar{c}_i = - \sum_{j \in J_k} \lambda_j a_j.$$

Y tomando $\lambda_j^k = 0$ para $j \notin J_k$ (las últimas restricciones del problema (P_{aux})) tenemos que se cumplen las condiciones de optimalidad descritas en el teorema 3.2.2, aplicado al problema (P_{aux}) . De donde se deduce que $x^k + d^k$ es la solución de dicho problema. \square

Para finalizar esta sección establecemos un resultado de convergencia para el algoritmo descrito.

Definición Se dice que \bar{x} es un *punto estacionario* si verifica la condición de optimalidad (3.5).

Teorema 4.2.3. *Si $\rho_k \neq 0$ en puntos estacionarios, entonces el algoritmo anterior encuentra la solución de (PCG) en un número finito de iteraciones.*

Demostración Sea \hat{x} la solución de PCG y \hat{J} su conjunto de trabajo asociado.

En primer lugar, notemos que si x^k es un punto estacionario generado por nuestro algoritmo y denotamos J_k a su conjunto de trabajo asociado, entonces x^k es la solución del problema¹

$$(PCG)_k \left\{ \begin{array}{l} \text{Min} \quad F(x) \\ \text{sujeto a} \quad a_i^T x = \tilde{b}_i \quad \text{para } i \in J_k, \\ \quad \quad \quad c_i^T x \leq b_i \quad \text{si } i \notin J_k. \end{array} \right.$$

Además usando la proposición 4.2.1 se tiene que

$$F(x^{k+1}) < F(x^k). \quad (4.9)$$

De donde se deduce que dados dos puntos estacionarios, x^k y x^t , generados por nuestro algoritmo con $k < t$ con sendos conjuntos de trabajo asociados, J_k y J_t , necesariamente $J_k \neq J_t$, de lo contrario tendríamos que $F(x^t) = F(x^k)$, llegando a una contradicción con (4.9).

Además, aparece al menos un punto estacionario cada n iteraciones del algoritmo. En efecto, basta notar que en cada conjunto de trabajo hay a lo sumo n restricciones y que cada iteración no asociada a un punto estacionario

¹Denotamos de nuevo por $\{a_j\}_{j \in J_k}$ las columnas de A_k . Así, a_j denotará \tilde{c}_j si la restricción está asociada a F o c_j si está asociada a K . Denotamos \tilde{b}_j el término independiente de esa restricción.

implica la adición de una restricción al conjunto de trabajo.

Por otro lado, existe tan sólo un número finito de conjuntos de trabajo posibles, lo que implica que en un número finito de iteraciones se obtendrá \hat{J} y nuestro algoritmo habrá alcanzado la solución. \square

4.3. Estudio de la degeneración

Diremos que en una iteración del algoritmo se produce *degeneración* si el desplazamiento calculado en (4.5) es 0 (en la práctica significa que el paso calculado es menor que la tolerancia). Esto implica que alguna restricción fuera del conjunto de trabajo nos impide avanzar en la dirección que hemos calculado. En la práctica si esto ocurre con las restricciones asociadas a F vamos a plantear una estrategia que tratará de evitar la degeneración. La idea tras esta estrategia es que al no ser una restricción en sentido estricto la que nos produce la degeneración, queremos saber si podemos hacer decrecer el valor del funcional manteniéndonos dentro del conjunto de puntos admisibles a costa de violar la restricción problemática, objetivo para el que calcularemos una nueva dirección de descenso. Para ello, supongamos que tenemos un punto x^k generado por nuestro algoritmo junto con un conjunto de trabajo J_k de matriz asociada A_k al comienzo de la iteración k y una restricción $\bar{c}_{i_0}^T x - \bar{b}_{i_0}$, con $1 \leq i_0 \leq m_{if} + m_{df}$ que nos produce degeneración.

Si a lo largo de la iteración se ha eliminado una restricción del conjunto de trabajo, $\tilde{a}_{j_0}^T x - \tilde{b}_{j_0}$, vamos a denotar \bar{J}_k y \bar{A}_k al conjunto de trabajo tras eliminar dicha restricción y a su correspondiente matriz asociada. Disponemos también de las matrices Z_k y \bar{Z}_k asociadas a los subespacios de los núcleos de A_k^T y \bar{A}_k^T respectivamente, de las que sabemos que $\bar{Z}_k = [Z_k \ z_0]$. Entonces tenemos dos posibilidades:

- I) Si $Z_k^T \bar{c}_{i_0} \neq 0$, entonces $\text{rango}([A_k \ \bar{c}_{i_0}]) > \text{rango}(A_k)$. En este caso se añade la restricción asociada a \bar{c}_{i_0} al conjunto de trabajo y no podemos evitar la degeneración.
- II) Si $Z_k^T \bar{c}_{i_0} = 0$, entonces $\text{rango}([A_k \ \bar{c}_{i_0}]) = \text{rango}(A_k)$. De donde:

- Existen $\beta_j \in \mathbb{R}$ tales que

$$\bar{c}_{i_0} = \sum_{j \in J_k} \beta_j a_j. \quad (4.10)$$

- Se deduce el hecho de haber eliminado una restricción (ya que $\bar{c}_{i_0}^T d^k \neq 0$), lo que implica que

$$\bar{g}^k = g^k + \sigma a_{j_0}.$$

con $\sigma \in \{-1, 0, 1\}$.

Para calcular una dirección de descenso, $\bar{d}^k = \bar{Z}_k \bar{d}_{\bar{Z}_k}$, violando la restricción que provoca degeneración tenemos que resolver el sistema de ecuaciones lineales

$$H_{\bar{Z}_k} \bar{d}_{\bar{Z}_k} = -\frac{1}{\alpha} \bar{Z}_k^T (\bar{g}^k + \delta \bar{c}_{i_0}) \quad \text{con } \delta \in \{-1, 1\}, \quad (4.11)$$

siendo $H_{\bar{Z}_k} = \bar{Z}_k^T H \bar{Z}_k$.

Veamos que el término independiente sólo tiene una coordenada distinta de cero. Como g^k pertenece al subespacio del rango de (A_k) , usando que $\bar{Z}_k = [Z_k \ z_0]$ y (4.10),

$$-\bar{Z}_k^T (\bar{g}^k + \delta \bar{c}_{i_0}) = \begin{pmatrix} 0 \\ -z_0^T (g^k + \sigma a_{j_0} + \delta \bar{c}_{i_0}) \end{pmatrix}. \quad (4.12)$$

Y aplicando (4.10) y (4.2) nos queda

$$-z_0^T (g^k + \sigma a_{j_0} + \delta \bar{c}_{i_0}) = z_0^T a_{j_0} (\lambda_{j_0}^k - \sigma - \delta \beta_{j_0}). \quad (4.13)$$

Ahora si $H_{\bar{Z}_k} = \bar{L}_k \bar{L}_k^T$ es la factorización de Cholesky del Hessiano reducido, usando (4.12) y (4.13) en (4.11), se tiene que

$$\bar{d}_{\bar{Z}_k} = \left(\frac{z_0^T a_{j_0} (\lambda_{j_0}^k - \sigma - \delta \beta_{j_0})}{\alpha l_{n_k, n_k}^2} \right) e_{n_k},$$

donde l_{n_k, n_k} es el último elemento diagonal de \bar{L}_k y e_{n_k} el n_k -ésimo vector de la base canónica de \mathbb{R}^{n_k} , siendo $n_k = n - m_k$ con m_k el número de restricciones en el conjunto de trabajo. Por tanto la dirección toma la forma $\bar{d}^k = \eta z_0$ con $\eta \in \mathbb{R}$. De donde, usando (4.10), la dirección verifica:

$$\bar{c}_{i_0}^T \bar{d}^k = \frac{\beta_{j_0} (z_0^T a_{j_0})^2 (\lambda_{j_0}^k - \sigma - \delta \beta_{j_0})}{\alpha l_{n_k, n_k}^2}, \quad (4.14)$$

$$a_{j_0}^T \bar{d}^k = \frac{(z_0^T a_{j_0})^2 (\lambda_{j_0} - \sigma - \delta \beta_{j_0})}{\alpha l_{n_k, n_k}^2}. \quad (4.15)$$

Las igualdades (4.14) y (4.15) nos ayudarán a determinar la existencia de una dirección de descenso admisible.

A partir de aquí distinguiremos tres casos según sea el tipo de la restricción eliminada.

- **Caso II.a:** La restricción eliminada ha sido de desigualdad de K . En este caso, la eliminación de la restricción no ha afectado al subgradiente por lo que $\sigma = 0$ y $\lambda_{j_0} < 0$. Además como queremos que se cumpla que $a_{j_0}^T \bar{d}^k < 0$ para continuar en la región de puntos admisibles, aplicando ahora (4.15) obtenemos

$$a_{j_0}^T \bar{d}^k < 0 \Leftrightarrow \lambda_{j_0} - \delta \beta_{j_0} < 0. \quad (4.16)$$

De nuevo tenemos que distinguir dos casos.

- **Caso II.a.1:** $\beta_{j_0} < 0$.

Usando (4.16) en (4.14) se tiene que $\bar{c}_{i_0}^T \bar{d}^k > 0$, luego $\delta = 1$, y por tanto la dirección será admisible si

$$\lambda_{j_0} < \beta_{j_0}.$$

- **Caso II.a.2:** $\beta_{j_0} > 0$.

Por (4.14) se tiene que $\bar{c}_{i_0}^T \bar{d}^k < 0$, luego $\delta = -1$ e $i_0 \leq m_{if}$. Por tanto la dirección será admisible si

$$\lambda_{j_0} + \beta_{j_0} < 0.$$

- **Caso II.b:** La restricción eliminada fue de igualdad asociada al funcional.

Así que haber eliminado la restricción implica que $\lambda_{j_0} \notin [-1, 1]$, luego tenemos dos nuevos casos.

- **Caso II.b.1:** $\lambda_{j_0} < -1$.

En este caso $\sigma = -1$ y por tanto, aplicando (4.15), tenemos la siguiente equivalencia para garantizar la admisibilidad

$$a_{j_0}^T \bar{d}^k < 0 \Leftrightarrow \lambda_{j_0} + 1 - \delta \beta_{j_0} < 0.$$

Si $\boxed{\beta_{j_0} < 0}$, entonces de (4.14), $\bar{c}_{i_0}^T \bar{d}^k > 0$, luego $\delta = 1$ y la dirección será admisible si

$$\boxed{\lambda_{j_0} + 1 < \beta_{j_0}.}$$

Si $\boxed{\beta_{j_0} > 0}$, entonces $\bar{c}_{i_0}^T \bar{d}^k < 0$, luego $\delta = -1$ e $i_0 \leq m_{if}$. Por tanto la dirección será admisible si

$$\boxed{\beta_{j_0} < -(\lambda_{j_0} + 1).}$$

• **Caso II.b.2:** $\boxed{\lambda_{j_0} > 1.}$

En este caso $\sigma = 1$ y la condición de admisibilidad viene dada por $a_{j_0}^T \bar{d}^k > 0$, o lo que es lo mismo, $\lambda_{j_0} - 1 - \delta\beta_{j_0} > 0$.

De forma análoga al caso II.b.1 se obtiene que:

Si $\boxed{\beta_{j_0} > 0}$ la dirección será admisible si

$$\boxed{\lambda_{j_0} - 1 > \beta_{j_0}.}$$

Y si $\boxed{\beta_{j_0} < 0}$, $\bar{c}_{i_0}^T \bar{d}^k < 0$, $\delta = -1$ e $i_0 \leq m_{if}$ y la dirección será admisible si

$$\boxed{\beta_{j_0} > 1 - \lambda_{j_0}.}$$

■ **Caso II.c:** La restricción eliminada fue de desigualdad asociada al funcional. Por tanto $\lambda_{j_0} \notin [0, 1]$, luego tenemos dos nuevos casos.

• **Caso II.c.1:** $\boxed{\lambda_{j_0} < 0.}$

En este caso $\sigma = 0$ y queremos que $a_{j_0}^T \bar{d}^k < 0$.

Aplicando (4.15) tenemos

$$a_{j_0}^T \bar{d}^k < 0 \Leftrightarrow \lambda_{j_0} - \delta\beta_{j_0} < 0$$

Si $\boxed{\beta_{j_0} < 0}$, entonces $\delta = 1$ y la dirección será admisible si

$$\boxed{\lambda_{j_0} < \beta_{j_0}.}$$

Si $\boxed{\beta_{j_0} > 0}$, entonces $\delta = -1$ e $i_0 \leq m_{if}$ y la dirección será admisible si

$$\boxed{\beta_{j_0} + \lambda_{j_0} < 0.}$$

• **Caso II.c.2:** $\lambda_{j_0} > 1$.

En este caso $\sigma = 1$ y $a_{j_0}^T \bar{d}^k > 0 \Leftrightarrow \lambda_{j_0} - 1 - \delta\beta_{j_0} > 0$.

De forma análoga a los casos anteriores se obtiene que:

Si $\beta_{j_0} > 0$ la dirección será admisible si

$$\lambda_{j_0} - 1 > \beta_{j_0}.$$

Y si $\beta_{j_0} < 0$, $\delta = -1$ e $i_0 \leq m_{if}$ y la dirección será admisible si

$$\beta_{j_0} > 1 - \lambda_{j_0}.$$

En caso de existir varias restricciones asociadas al funcional que impidan avanzar en la dirección calculada, estudiaremos la existencia de dirección para cada una de ellas.

Capítulo 5

Recuperación de imágenes

5.1. Modelización del problema como (PCG)

Antes de comenzar, debemos definir claramente lo que es una imagen y a lo que nos referimos con recuperarla.

Definición Una *imagen en escala de grises* es una aplicación

$$\begin{aligned} \omega : \Omega = [0, 1] \times [0, 1] &\longrightarrow [0, +\infty) \\ (x, y) &\mapsto f(x, y) \end{aligned}$$

donde se asocia a cada punto (x, y) del cuadrado unidad un número $\omega(x, y)$ que determinará la intensidad de brillo o escala de gris de la imagen en ese punto.

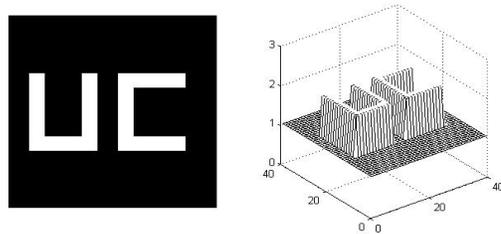


Figura 1: Ejemplo de una imagen y su gráfica en escala de grises

Por diversos motivos, una imagen ω puede verse afectada por dos tipos de distorsión, un *difuminado* y una distorsión aditiva o *ruido* de forma que se tiene:

$$z = H\omega + \eta,$$

donde z es la imagen distorsionada, ω es la imagen original, H es el difuminado y η el ruido. Nuestro objetivo es entonces reconstruir ω a partir de z en el caso especial de ausencia de difuminado, es decir, $z = \omega + \eta$. Por ejemplo, en la figura 1 tenemos una *imagen* a la izquierda y su correspondiente escala de grises o intensidades (ω) a la derecha, y en la figura 2 tenemos ambas representaciones (z) para una distorsión aditiva de la imagen de la figura 1 con la que hemos trabajado.

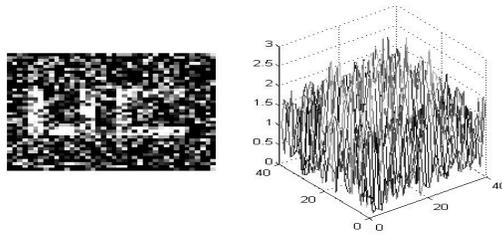


Figura 2: Ejemplo de una imagen distorsionada y su escala de grises.

La eliminación o reducción de este tipo de ruido es fundamental en diversos procesos de visión por computador, por ejemplo en imágenes tomadas con una cámara digital u otros dispositivos tales como un escáner. En algunas aplicaciones (por ejemplo médicas) es esencial que el tratamiento de las imágenes conserve los bordes, ya que son elementos críticos en etapas posteriores para el conteo de objetos o el reconocimiento de estructuras.

Si observamos la imagen distorsionada de la izquierda de la figura 2 fácilmente reconoceremos las siglas UC, lo que nos puede llevar a pensar que la recuperación planteada es un proceso en cierto sentido trivial o innecesario, sin embargo basta un vistazo a su escala de grises para darnos cuenta de que esto no es así. Este efecto se debe a que la visión humana también posee algún tipo de procesamiento de imágenes, de forma que podemos reconocer ciertos patrones.

Para abordar el problema de la recuperación de imágenes, teselaremos el dominio de la imagen utilizando $n = m^2$ cuadrados o *píxeles* y supondremos

constante la intensidad en cada uno de los subdominios:

$x_{1,m}$	$x_{2,m}$	\cdots	$x_{m,m}$
\vdots	\vdots	\dots	\vdots
$x_{1,2}$	$x_{2,2}$	\cdots	$x_{m,2}$
$x_{1,1}$	$x_{2,1}$	\cdots	$x_{m,1}$

donde $x_{i,j}$ representa la intensidad en el cuadrado (i, j) .

Así que la imagen distorsionada vendrá dada por una matriz de datos, z , que reordenada en forma vectorial da lugar al problema de mínimos cuadrados regularizado:

$$(\mathcal{PCG}_{\mathcal{I}}) \begin{cases} \text{Min } F(x) = \frac{\alpha}{2} \|x - z\|_2^2 + TV(x) \\ \text{sujeto a } x \geq 0, \end{cases}$$

donde el parámetro de penalización, α , mide la prioridad que damos al término mínimo-cuadrático frente a

$$TV(x) = \frac{1}{m} \left(\sum_{i=1}^m \sum_{j=1}^{m-1} |x_{i,j} - x_{i,j+1}| + \sum_{i=1}^{m-1} \sum_{j=1}^m |x_{i,j} - x_{i+1,j}| \right).$$

Este segundo término, $TV(x)$, mide la variación total en niveles de gris de una imagen x reordenada en forma matricial¹. Su uso está especialmente indicado para recuperar imágenes con bordes cortantes (ver por ejemplo [2] y [5]). Además puede expresarse como la norma de un producto matriz-vector. En efecto, sea $E = (e_{i,j})$ la matriz $(m-1) \times m$ dada por

$$e_{i,j} = \begin{cases} 1 & \text{si } i = j, \\ -1 & \text{si } i = j - 1, \\ 0 & \text{en otro caso.} \end{cases} \quad \text{para } i = 1, \dots, m-1,$$

Entonces

$$TV(x) = \|\bar{C}^T x\| \quad \text{con} \quad \bar{C} = \frac{1}{m} \begin{pmatrix} E \otimes_K I \\ I \otimes_K E \end{pmatrix},$$

¹Utilizaremos la misma notación para referirnos a una imagen tanto si sus valores están dados por una matriz como si están dados en forma vectorial.

donde \otimes_K es el producto de Kronecker y \bar{C} tiene $m_{if} = 2m(m-1)$ columnas y m^2 filas. De este modo el problema $(\mathcal{PCG}_{\mathcal{I}})$ puede reescribirse como

$$(\mathcal{PCG}_{\mathcal{I}}) \left\{ \begin{array}{l} \text{Min } F(x) = \frac{\alpha}{2} \|x - z\|_2^2 + \|\bar{C}x\|_1 \\ \text{sujeto a} \quad \quad \quad x \geq 0. \end{array} \right.$$

El término de regularización, $\|\bar{C}x\|_1$, se ha usado en otras formulaciones para la recuperación de imágenes, como por ejemplo en [4]. Otras regularizaciones han sido utilizadas por ejemplo en [3] y [5].

Notemos que en nuestra formulación hemos añadido restricciones de cota sobre las variables, pero podemos encontrar otras formulaciones que han eliminado tales restricciones. Esto se debe fundamentalmente a que la resolución del problema es más sencilla.

Para la resolución numérica de $(\mathcal{PCG}_{\mathcal{I}})$ podemos considerar una adaptación del algoritmo de conjunto activo presentado en la sección 4.1 teniendo en cuenta que en este caso $H = I$, lo que conlleva algunas simplificaciones como por ejemplo el cálculo de la dirección de descenso (que no requiere la factorización de Cholesky de H_Z para resolver (4.4)).

5.2. Otra formulación equivalente

El problema de la recuperación de imágenes, como es usual en matemáticas, puede enfocarse desde varias perspectivas distintas. Ahora consideraremos una formulación equivalente a $(\mathcal{PCG}_{\mathcal{I}})$, aumentando considerablemente el número de variables, para obtener un problema de programación cuadrática convexa estándar (sin términos no diferenciables). Introduciendo variables auxiliares, podemos convertir el problema $(\mathcal{PCG}_{\mathcal{I}})$ en el problema

$$(\mathcal{PC}_{\mathcal{I}}) \left\{ \begin{array}{l} \text{Min } G(x, u, v) = \frac{\alpha}{2} \|x - z\|_2^2 + \sum_{i=1}^{m_{if}} (u_i + v_i) \\ (x, u, v) \in \mathbb{R}^n \times \mathbb{R}^{m_{if}} \times \mathbb{R}^{m_{if}} \\ \text{sujeto a} \quad \quad \quad \bar{C}^T x - u + v = 0, \\ \quad \quad \quad \quad \quad \quad \quad \quad x, u, v \geq 0. \end{array} \right.$$

Veamos que ambos problemas son equivalentes. Antes de comenzar, llamemos \hat{p} el vector de términos lineales de $(\mathcal{PCG}_{\mathcal{I}})$ y \tilde{p} el de $\mathcal{PC}_{\mathcal{I}}$, de forma que se

tiene:

$$\tilde{p} = \begin{pmatrix} \hat{p} \\ e \\ e \end{pmatrix},$$

siendo $e = (1, \dots, 1)^T$. En primer lugar vamos a enunciar las condiciones de optimalidad de ambos problemas. Aplicando el teorema 3.2.2 tenemos que \bar{x} es solución de $\mathcal{PCG}_{\mathcal{I}}$ si, y sólo si existen $\bar{\lambda} \in \mathbb{R}^t$ y $\bar{\mu} \in \mathbb{R}^n$ tales que:

$$\alpha \bar{x} + \hat{p} + \bar{C} \bar{\lambda} - \bar{\mu} = 0, \quad (5.1)$$

$$\bar{\lambda}_i = \begin{cases} 1 & \text{si } \bar{c}_i^T \bar{x} > 0, \\ -1 & \text{si } \bar{c}_i^T \bar{x} < 0, \end{cases} \quad (5.2)$$

$$\bar{\lambda}_i \in [-1, 1] \quad \text{si } \bar{c}_i^T \bar{x} = 0, \quad (5.3)$$

$$\bar{\mu} \geq 0, \quad \bar{\mu} \cdot \bar{x} = 0, \quad (5.4)$$

$$\bar{x} \geq 0. \quad (5.5)$$

2

Por otra parte, $(\tilde{x}, \tilde{u}, \tilde{v})^T$ es solución de $(\mathcal{PC}_{\mathcal{I}})$ si y sólo si existen $\tilde{\lambda} \in \mathbb{R}^{m_{if}}$, $\tilde{\mu}_x \in \mathbb{R}^n$, $\tilde{\mu}_u \in \mathbb{R}^{m_{if}}$ y $\tilde{\mu}_v \in \mathbb{R}^{m_{if}}$ cumpliendo:

$$\begin{pmatrix} \alpha \tilde{x} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{p} \\ e \\ e \end{pmatrix} + \begin{pmatrix} \bar{C} \\ -I \\ I \end{pmatrix} \tilde{\lambda} - \begin{pmatrix} \tilde{\mu}_x \\ \tilde{\mu}_u \\ \tilde{\mu}_v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.6)$$

$$\left. \begin{aligned} \tilde{\mu}_x \cdot \tilde{x} &= 0, & \tilde{\mu}_x &\geq 0, \\ \tilde{\mu}_u \cdot \tilde{u} &= 0, & \tilde{\mu}_u &\geq 0, \\ \tilde{\mu}_v \cdot \tilde{v} &= 0, & \tilde{\mu}_v &\geq 0, \end{aligned} \right\} \quad (5.7)$$

$$\left. \begin{aligned} \bar{C}^T \tilde{x} - \tilde{u} + \tilde{v} &= 0, \\ \tilde{x}, \tilde{u}, \tilde{v} &\geq 0. \end{aligned} \right\} \quad (5.8)$$

Ahora veamos la equivalencia de estos dos sistemas de optimalidad. Supongamos que $(\tilde{x}, \tilde{u}, \tilde{v})$ es solución de $(\mathcal{PC}_{\mathcal{I}})$, entonces de la primera igualdad de la condición (5.6) obtenemos claramente la condición (5.1) y de las otras dos igualdades tenemos que:

$$\tilde{\lambda} = e - \tilde{\mu}_u, \quad (5.9)$$

$$\tilde{\lambda} = -e + \tilde{\mu}_v, \quad (5.10)$$

²Dados dos vectores, $u, v \in \mathbb{R}^n$, denotaremos $u \cdot v$ al vector de \mathbb{R}^n cuyas coordenada i -ésima es $u_i v_i$.

de donde usando la positividad de los multiplicadores (ver (5.7)) se deduce que $\tilde{\lambda}_i \in [-1, 1]$. Además, si $\bar{c}_i^T \tilde{x} > 0$, por (5.8) tenemos que $\tilde{u}_i - \tilde{v}_i > 0$, luego $\tilde{u}_i > \tilde{v}_i \geq 0$, lo que, por (5.7), implica $(\tilde{\mu}_u)_i = 0$ y por tanto, de (5.10), $\tilde{\lambda}_i = 1$. De forma análoga se obtiene que si $\bar{c}_i^T \tilde{x} < 0$, entonces $\tilde{\lambda}_i = -1$. Por todo ello, obtenemos las condiciones (5.2) y (5.3). Por último, es claro que tomando $\bar{x} = \tilde{x}$ y $\bar{\mu} = \tilde{\mu}_x$ se satisfacen las condiciones (5.4) y (5.5), luego \tilde{x} es la solución de (\mathcal{PCG}_I) .

Ahora, para el recíproco, si \bar{x} es la solución de (\mathcal{PCG}_I) , tomando $\tilde{u}_i = \max \{\bar{c}_i^T \bar{x}, 0\}$ y $\tilde{v}_i = \max \{-\bar{c}_i^T \bar{x}, 0\}$ se cumple la condición (5.8). Con $\tilde{\mu}_x = \bar{\mu}$, $\tilde{\mu}_u = e - \bar{\lambda}$ y $\tilde{\mu}_v = e + \bar{\lambda}$, es claro que se cumplen la condición (5.6), las condiciones sobre el signo y la primera igualdad de (5.7). Veamos que estos valores también cumplen el resto de la condición (5.7), distinguiendo tres casos:

- Si $\bar{c}_i^T \bar{x} > 0$, entonces $\bar{\lambda}_i = 1$, $(\tilde{\mu}_u)_i = 0$ y $\tilde{v}_i = 0$.
- Del mismo modo, si $\bar{c}_i^T \bar{x} < 0$ se tiene que $\bar{\lambda}_i = -1$, por tanto $(\tilde{\mu}_v)_i = 0$ y $\tilde{u}_i = 0$.
- Por último, si $\bar{c}_i^T \bar{x} = 0$ tenemos $\tilde{u}_i = 0$ y $\tilde{v}_i = 0$.

Luego se cumple la condición (5.7) completa, obteniéndose que $(\bar{x}, \tilde{u}, \tilde{v})$ cumple las condiciones de optimalidad para (\mathcal{PC}_I) y por tanto es la solución de ese problema. \square

Hemos logrado obtener una formulación cuadrática estándar equivalente a (\mathcal{PCG}_I) , pero aumentando considerablemente el número de variables (ver Tabla 1). En principio es más sencillo abordar la resolución numérica de (\mathcal{PC}_I) , pero su talla nos obliga a utilizar algoritmos de puntos interiores como por ejemplo el implementado en la función *quadprog* de MATLAB que describimos brevemente a continuación (para más detalles ver [9]).

Tamaño de imagen	Nº de variables en (\mathcal{PCG}_I)	Nº de variables en (\mathcal{PC}_I)
10 × 10	100	460
50 × 50	2500	12300
100 × 100	10000	49600

Tabla 1: Comparativa del número de variables para ambas formulaciones.

Por simplicidad, consideraremos la siguiente formulación

$$\left\{ \begin{array}{l} \text{Min} \quad \frac{1}{2}y^T \tilde{H}y + \tilde{p}^T y \\ \text{sujeto a} \quad y \in \mathbb{R}^n, \\ \quad \quad \quad A_{eq}^T y = b_{eq}, \\ \quad \quad \quad y \geq 0 \end{array} \right.$$

que contiene a (\mathcal{PC}_T) como caso particular. Así, las condiciones (5.6)-(5.8) pueden escribirse como:

$$\tilde{H}\tilde{y} + \tilde{p} + A_{eq}\tilde{\lambda} - \tilde{\mu} = 0, \quad (5.11)$$

$$A_{eq}\tilde{y} - b_{eq} = 0, \quad (5.12)$$

$$\tilde{y} \cdot \tilde{\mu} = 0, \quad (5.13)$$

$$\tilde{y}, \tilde{\mu} \geq 0, \quad (5.14)$$

y la idea del método es resolver este sistema de ecuaciones no lineales con restricciones sobre el signo de las variables. Para ello, se aplica una variante del método de Newton-Rhapson, ya que la presencia de las inecuaciones (5.14) hace que el algoritmo original en ocasiones produzca direcciones no admisibles o que lleven asociado un paso demasiado pequeño. Para tratar esa variante, consideremos en lugar de (5.11)-(5.13), el siguiente sistema perturbado

$$\begin{pmatrix} \tilde{H}y + \tilde{p} + A_{eq}\lambda - \mu \\ A_{eq}y - b_{eq} \\ y \cdot \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \tau e \end{pmatrix},$$

con $\tau > 0^3$ y la dirección de Newton asociada, (d_y, d_λ, d_μ) , que verifica el sistema de ecuaciones lineales

$$\begin{pmatrix} \tilde{H} & A_{eq} & -I \\ A_{eq} & 0 & 0 \\ \Lambda & 0 & Y \end{pmatrix} \begin{pmatrix} d_y \\ d_\lambda \\ d_\mu \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ -y \cdot \mu + \sigma \tau e \end{pmatrix}, \quad (5.15)$$

donde Λ e Y son matrices diagonales cuyas entradas no nulas son $\tilde{\mu}$ e y respectivamente, r_1 y r_2 son los residuos de los grupos de ecuaciones (5.11) y (5.12) y $\sigma \in [0, 1]$. En realidad, en cada iteración se resuelve (5.15) con $\sigma = 1$

³ τ es el parámetro que define el camino central, \mathcal{C} , en el método de puntos interiores.

y con $\sigma = 0$. Si $\sigma = 1$ tenemos una *dirección correctora* hacia el camino central y con $\sigma = 0$ obtenemos una *dirección predictora*, que es la dirección estándar del método de Newton.

La principal diferencia entre los algoritmos de conjunto activo y los algoritmos de puntos interiores reside en que los primeros calculan candidatos a multiplicadores que cumplen siempre la ecuación no lineal (5.13) de las condiciones de optimalidad e iteran hasta cumplir la condición sobre el signo de los multiplicadores. Por el contrario los algoritmos de puntos interiores mantienen siempre multiplicadores con el signo correcto, y tratan de disminuir el residuo de la ecuación no lineal (5.13) en cada iteración.

5.3. Ensayos numéricos

A continuación mostramos los resultados numéricos obtenidos con nuestro algoritmo para dos imágenes de 40×40 píxeles. Las figuras 3 y 4 corresponden a la primera de ellas y las figuras 5 y 6 a la segunda. En todas las figuras a la izquierda se presenta la imagen original (ω), en el centro la imagen perturbada o dato (z), que hemos construido introduciendo un ruido aleatorio en el intervalo $[-1, 1]$ en cada píxel, permitiendo así que el ruido alcance un 50 % del valor máximo de la escala original, y a la derecha mostramos la imagen recuperada (\bar{x}). En las figuras pares se presentan las escalas de grises de las correspondientes imágenes de las figuras impares. También incluimos algunas tablas que muestran datos relevantes en cuanto al coste y a la calidad de las soluciones obtenidas tanto con nuestro código como con el código *quadprog* de MATLAB. Este último nos permite trabajar con imágenes más grandes, a modo de ejemplo presentamos un texto obtenido de la página web del libro [6] y una señal de tráfico (ver figuras 7-10).⁴

En las tablas se utiliza la notación:

- *CA1/CA2*: Algoritmo de conjunto activo con la estrategia 1 o 2 respectivamente (ver sección 4.1).
- *PI*: Algoritmo de puntos interiores implementado en *quadprog*.
- *ICD*: Número de iteraciones con degeneración.

⁴Todos los ensayos se han llevado a cabo en un Pentium(R) Dual-Core, 64 bits, 2.80 GHz utilizando Windows 7 como sistema operativo y la version R2011B de MATLAB.

- *EMD*: Error medio de los datos, $EMD(z) = \sum_{i=1}^n \frac{|z_i - \omega_i|}{n}$.
- *EMXD*: Error máximo de los datos.
- *EMS*: Error medio de la solución en relación a ω .
- *EMXS*: Error máximo de la solución.
- *SMI*: Superado el máximo de iteraciones fijado.
- *TOL*: Tolerancia para el criterio de parada⁵.

En la tabla 2 evaluamos, para cada ensayo (asociado a una imagen y a uno de los dos códigos), el coste de la imagen recuperada medido en tiempo de ejecución y en número de iteraciones.

Ensayo	Imagen	Código	α	Tiempo	Iteraciones	ICD
1	Cruz	CA1	6.25e-2	9 min 43 s	2000	524
2	Cruz	CA2	6.25e-2	11 min 49 s s	2000	343
3	Cruz	PI	6.25e-2	1 s	12	-
4	Logo UC	CA1	5.00e-2	10 min 1 s	2000	463
5	Logo UC	CA2	5.00e-2	12 min 11 s	2000	370
6	Logo UC	PI	5.00e-2	1 s	12	-
7	Texto	PI	1.07e-2	1 min 23 s	13	-
8	Señal	PI	1.20e-2	58 s	13	-

Tabla 2: Evaluación del coste de algunos ensayos.

Comparando las dos últimas columnas de la tabla 2 podemos observar el gran número de iteraciones en las que se produce degeneración con el algoritmo de conjunto activo (CA), que oscilan entre el 17% y el 26% del total. Esto se debe a que las dos primeras imágenes, aunque en apariencia son sencillas, en realidad son propensas a producir degeneración ya que contienen muchas restricciones activas linealmente dependientes debido a sus grandes áreas monocromáticas.

En la tabla 3 se analizan las calidades de las imágenes obtenidas en los

⁵En nuestro código, la tolerancia se utiliza para comprobar las condiciones del teorema 3.2.2, mientras que en quadprog se utiliza para comprobar las condiciones (5.11)-(5.13)

Ensayo	ICD	EMD	EMXD	EMS	EMXS	Parada	TOL
1	524	0.4949	0.9993	0.1326	0.8456	SMI	10^{-12}
2	343	0.4949	0.9993	0.1232	0.8353	SMI	10^{-12}
3	-	0.4949	0.9993	0.1223	0.8351	Satisfactoria	10^{-8}
4	492	0.4907	0.9999	0.1172	0.8110	SMI	10^{-12}
5	378	0.4907	0.9999	0.1153	0.7668	SMI	10^{-12}
6	-	0.4907	0.9999	0.1148	0.7578	Satisfactoria	10^{-8}
7	-	0.5010	1.0000	0.1214	0.9974	Satisfactoria	10^{-8}
8	-	0.4994	1.0000	0.0717	0.9069	Satisfactoria	10^{-8}

Tabla 3: Calidades de las imágenes de los datos y de las recuperadas.

Ensayo	EMS	EMXS	Iteraciones	Tiempo
1	0.1326	0.8456	1477	6 min 49 s
2	0.1326	0.8456	1477	6 min 45 s
4	0.1172	0.8110	1538	6 min 19 s
5	0.1172	0.8110	1538	6 min 18 s

Tabla 4: Resultados obtenidos sin tratar la degeneración

ensayos de la tabla 2. La medición de los errores se ha hecho en relación a la imagen original, que notemos no es la solución del problema de optimización planteado. Por otro lado vemos que aunque la reducción del error máximo no es demasiado buena, la reducción en el error medio alcanza el 85 %.

En la tabla 4 vemos los resultados obtenidos si paramos la ejecución de nuestro código en el momento que aparece la primera iteración con degeneración. En primer lugar se observa que tanto los tiempos de ejecución como el número de iteraciones se reducen drásticamente y obtenemos resultados similares a los obtenidos cuando utilizamos el proceso completo, motivo por el que no se representan las soluciones asociadas a esta tabla. Por otro lado se pone de manifiesto que la elección de la estrategia (CA1 o CA2) es influyente únicamente en la presencia de degeneración.

Debido a que las calidades de las soluciones obtenidas con nuestro código (CA1 y CA2) son similares a las de las soluciones obtenidas con quadprog (PI), en las figuras sólo presentamos un ensayo de cada imagen.

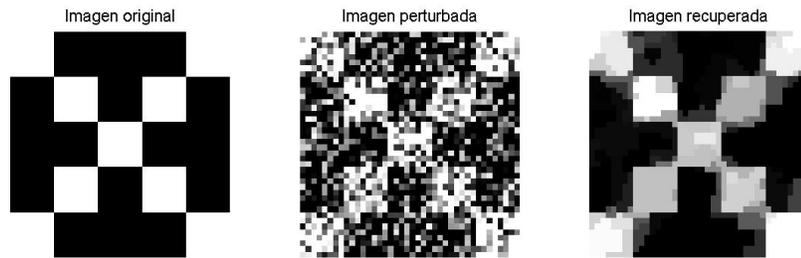


Figura 3: Imágenes relativas al ensayo 2

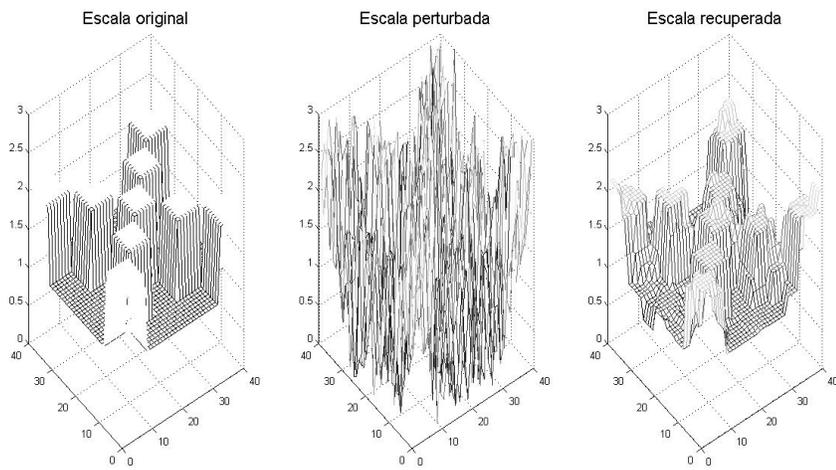


Figura 4: Escalas de grises de las imágenes en la figura 3

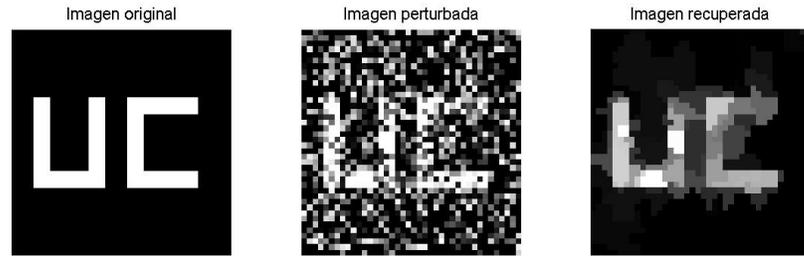


Figura 5: Imágenes relativas al ensayo 4

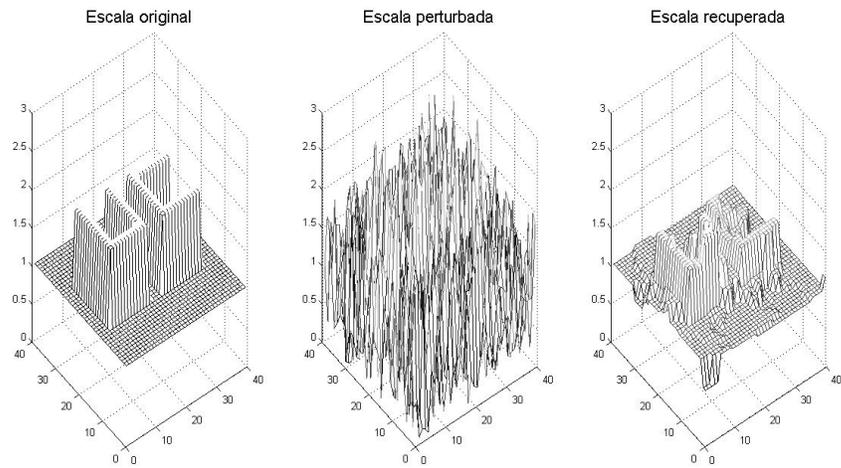


Figura 6: Escalas de grises de las imágenes de la figura 5



Figura 7: Imágenes relativas al ensayo 7

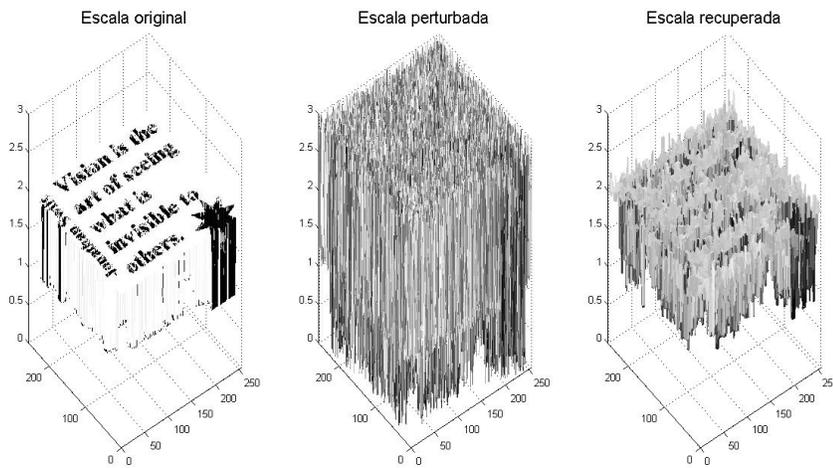


Figura 8: Escalas de grises de las imágenes de la figura 7

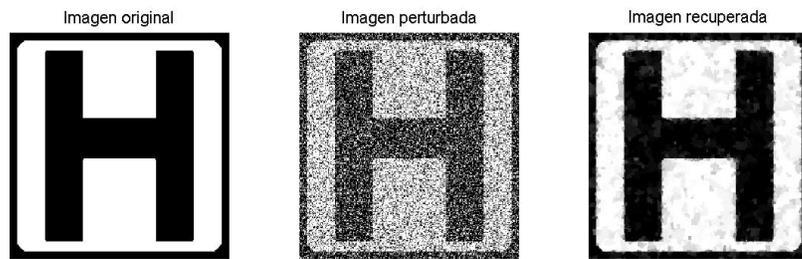


Figura 9: Imágenes relativas al ensayo 8

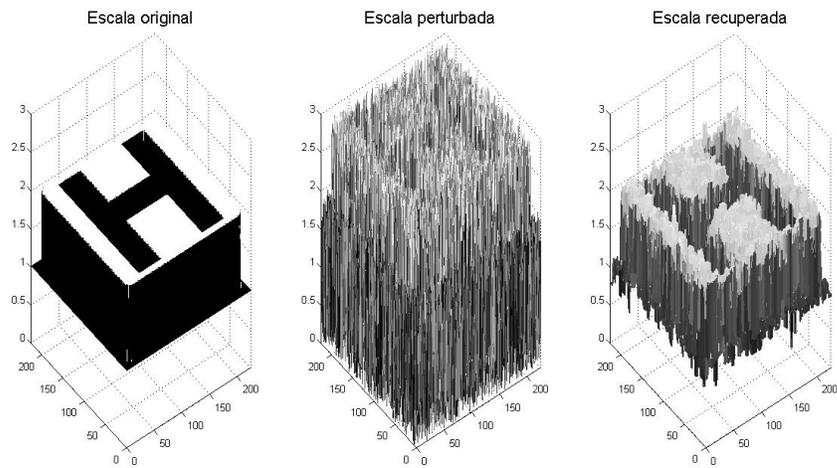


Figura 10: Escalas de grises de las imágenes de la figura 9

Conclusiones

En este trabajo hemos partido de un problema conocido para un estudiante del Grado en Matemáticas de la Universidad de Cantabria (minimizar una función cuadrática sujeta a un conjunto de restricciones lineales) y le hemos añadido términos convexos no diferenciables (asociadas a la función máximo y a la función valor absoluto), lo cual nos ha llevado al estudio del cálculo subdiferencial de funciones convexas para poder determinar unas condiciones de optimalidad para nuestro problema (*PCG*) (teorema 3.2.2) que son una extensión del teorema de los multiplicadores de Lagrange para programación cuadrática estándar.

Para la resolución numérica del problema (*PCG*) hemos estudiado e implementado el algoritmo propuesto en [7], así como una nueva estrategia para el tratamiento de la degeneración debido a la fuerte presencia de estas situaciones en los problemas de recuperación de imágenes que hemos abordado.

En la experimentación numérica con los problemas de recuperación de imágenes, hemos logrado calidades similares en las imágenes recuperadas utilizando nuestro código y el código *quadprog* de MATLAB, sin embargo observamos que nuestro algoritmo es más lento, debido a que nuestro código sigue una estrategia de conjunto activo usando la factorización QR frente a *quadprog* que sigue un método de puntos interiores con las factorizaciones adaptadas para matrices huecas. El uso de *quadprog* precisa una reformulación del problema (*PCG*) para obtener un problema diferenciable lo que supone aumentar el número de variables en $4\sqrt{n}(\sqrt{n}-1)$, siendo n el número de variables de (*PCG*). Así que en otras situaciones en las que las matrices no sean huecas, nuestro código puede resultar ventajoso. En cuanto a los problemas presentados observamos que, aunque no se cumplen las condiciones de optimalidad con la tolerancia deseada, las imágenes recuperadas pueden considerarse satisfactorias desde el punto de vista de la eliminación del ruido.

Finalmente debemos apuntar que la técnica de optimización utilizada solamente permite una mejora en la delimitación de los grandes cambios de intensidad, por lo que sería necesario un post-procesamiento de la imagen solución obtenida para recuperar otros detalles.

Glosario

A continuación describimos algunas de las notaciones que hemos utilizado a lo largo del trabajo:

- $\mathbb{R}^+ = \{x \in \mathbb{R} : x > 0\}$.
- $x \geq 0$ con $x \in \mathbb{R}^n$ denota $x_i \geq 0$ para $i = 1, \dots, n$.
- $\langle x, y \rangle = x^T y$.
- $x + S$ con $x \in \mathbb{R}^n$ y $S \subset \mathbb{R}^n$ denota el conjunto $\{x + s : s \in S\}$.
- $[A \ B]$ con $A \in \mathbb{R}^{n \times m_1}$, $B \in \mathbb{R}^{n \times m_2}$ denota la matriz $n \times (m_1 + m_2)$ cuyas m_1 primeras columnas son las columnas de A y las m_2 últimas son las columnas de B .
- AS con $A \in \mathbb{R}^{m \times n}$ y $S \subset \mathbb{R}^n$ denota el conjunto $\{a_i s : s \in S\}$, siendo a_i las columnas de A .

Bibliografía

- [1] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, 2003.
- [2] E. Casas, K.Kunisch, and C. Pola. Regularization by functions of bounded variation and applications to image enhancement. *Applied Mathematics and Optimization*, 40:229–257, 1999.
- [3] B. Kim. *Numerical Optimization Methods for Imagen Restoration*. PhD thesis, Department of Management Science and Engineering, Stanford University, 2002.
- [4] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Trans. Image Process.*, 5:987–995, 1996.
- [5] K. Majava. *Optimization-Based Techniques for Imagen Restoration*. PhD thesis, Department of Mathematical Information Technology, University of Jyväskylä, 2001.
- [6] D. P. O’Learly. *Scientific Computing with Case Studies*. SIAM, 2009.
- [7] C. Pola. *Algoritmos Numéricos para la resolución de problemas de optimización con restricciones*. PhD thesis, Dpto. Matemáticas, Estadística y Computación, Universidad de Cantabria (Spain), 1992.
- [8] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1972.
- [9] S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, 1997.