FACULTAD DE MATEMÁTICAS

DEPARTAMENTO DE ANÁLISIS MATEMÁTICO

Trabajo Fin de Grado

MÉTODOS DE OPTIMIZACIÓN CONVEXA PARA RESOLVER PROBLEMAS EN APRENDIZAJE AUTOMÁTICO

(Convex Optimization Methods for Machine Learning)

Mateo Carmona Carrasco

Dirigido por:
Dra. Dña. Victoria Martín Márquez

Sevilla, Septiembre 2021.

Resumen

Nowadays, the learning methods developed to solve optimization problems turn out to have strange behavior when we work analyzing data of a large dimension, mainly developed in a statistical context. Many of these methods have become computationally hard and slow. Right, the benefits of methods optimization only come when the problem is kwnown ahead of time to be convex. Methods, such as gradient descent or geometric descent, are capable of dealing with these problems efficiently, thus obtaining great results. Throughout this work, we will also expose and develop other methods derived from machine learning, thus concluding with the bootstrap model, which has now became relevant.

Índice general

Capítulo 1

Introducción

El aprendizaje automático o como es mundialmente conocido *"machine learning"* ha ido evolucionando con el paso de los años desde que, en 1950, *Alan Turing* crease el *"test de Turing"* para determinar si una máquina era realmente inteligente: dicho test consistía en que la máquina tenía que ser capaz de engañar a un humano haciéndole creer que era humana en lugar de un ordenador. Bien es cierto que no todo fue un camino de rosas, y desde entonces hasta el día de hoy ha sufrido varios estancamientos (el primero de ellos a mitad de la década de los 70 y otro a finales de los 80) pero siempre ha sabido cómo reinventarse gracias a la capacidad de almacenar cada vez mayor dimensión de datos y cómo gestionarlos. Su relevancia viene a raíz de que se toma como herramienta fundamental en multitud de sectores, tales como la Medicina (los sistemas que lo incorporan pueden "aprender" cuándo se dan las condiciones para que un paciente sufra una enfermedad), Finanzas (para la detección de fraudes financieros), Marketing (análisis de opiniones), Recursos Humanos (crear descripciones de puestos de trabajo que son neutrales en cuanto al género para atraer a los mejores candidatos posibles, ya sean hombres o mujeres), Robótica (el aumento del tiempo de actividad y la productividad del mantenimiento predictivo), entre muchas otras. Es decir, el aprendizaje automático engloba todo tipo de métodos capaces de predecir el comportamiento más probable a partir de un gran volumen de datos. Por tanto, la finalidad no es más que conseguir siempre ésto de la mejor manera posible, por lo que es irremediable la aparición de un problema de optimización: es decir, este tipo de problemas desempeñan una labor fundamental en el campo del aprendizaje automático. Además, con el paso de los años, los volúmenes de datos a almacenar y gestionar han ido aumentando por lo que se hace necesario que los algoritmos para resolver estos problemas tengan buenas propiedades, principalmente, que tengan una rápida resolución y una gran habilidad para reaccionar y adaptarse sin perder calidad (a parte de tener otras otras de interés como pueden ser la estabilidad...).

He aquí la razón fundamental por la que contemplaremos en nuestro trabajo los problemas de optimización convexa, ya que éstos son más rápidos a la hora de encontrar una solución, más simples y tienen un coste computacional menor. Bien es cierto que existen otros problemas de optimización que se asemejan más a la realidad estudiada pero la mayoría de ellos suelen ser irresolubles; por lo que a priori, los problemas de optimización convexa se nos presentan con mayor facilidad para poder tratarlos.

En este trabajo vamos a analizar y estudiar algunos métodos de optimización

convexa y posibles variaciones que surgen de algunos de ellos y finalizaremos con el problema LASSO cuyo punto álgido es visto mediante el modelo *bootstrap*. Comenzaremos con una serie de nociones previas, tales como en qué consiste la optimización convexa, para dar paso a los métodos de descenso por gradiente, uno de los algoritmos más populares en el campo del aprendizaje automático, particularmente por su uso extensivo en el campo de las redes neuronales. Estos algoritmos parten de una serie de hipótesis poco restrictivas sobre la función objetivo, las cuales suelen ser en su mayoría funciones diferenciables pero, se pueden extender a funciones no diferenciables haciendo uso del concepto de subgradiente; una de las ventajas que nos va a aportar el trato con funciones convexas es que nos van a garantizar de que, en el caso de existencia de mínimo, este va a ser único, por lo que una vez encontrado el minimizante, pararemos el algoritmo. Todo esto a simple vista parece idealizar tal tipo de algoritmo pero de no hacer una correcta elección de la longitud de paso puede tener como consecuencia la divergencia del método o conducirnos a soluciones no aptas. Por ello, una vez dado introducción a dicho método, nos centraremos en una buena elección de la longitud de paso.

Resumiendo, comenzaremos con una serie de nociones básicas sobre convexidad y su relevancia, pero a su vez contextualizaremos el marco inicial donde nos encontraremos a la hora de tratar con los problemas de optimización convexa, tales como el subconjunto de partida. A continuación, veremos el método de descenso por gradiente, imponiendo ciertas hipótesis sobre la función objetivo, principalmente la convexidad y una correcta elección de la longitud de paso, y posteriormente daremos alguno de los métodos alternativos que surgen del estudio previo de éste. Por último, introduciremos el problema LASSO concluyendo como forma resolutiva de éste con el bootstrap, que como veremos, no es más que una "técnica de remuestreo" que se emplea en estadística cada vez con más frecuencia gracias a la potencia de los ordenadores actuales, que permiten hacer cálculos que antes podían ser inconcebibles.

Capítulo 2

Preliminares

En este breve capítulo daremos una serie de notaciones que vamos a emplear a lo largo del trabajo para evitar confusiones y tener un contexto previo de las diferentes notaciones antes de analizar el contenido.

Principalmente, aunque de ser lo contrario lo especificaremos, trabajaremos en todo el espacio $\mathbb{R}^n$. Un elemento cualquiera de dicho espacio será x, el cual constará de n componentes, es decir, $x = (x_1, x_2, ..., x_n)$ o bien $x(i)$, $i = 1, ..., n$. Además, trabajaremos con matrices, donde un elemento será $A \in \mathbb{R}^n \times \mathbb{R}^n$, de manera que el elemento que ocupe la fila i y la columna j lo denotaremos como $a(i, j)$. Paralelamente también podremos referirnos a la fila i-ésima como $A(i, :)$ e igualmente a la columna j-ésima como $A(:, j)$. Bien es cierto que en el capítulo 4 nos referiremos al conjunto

$$S = \Big\{ x \in \text{dom } f \mid f(x) \leq f(x_0) \Big\} \tag{2.1}$$

donde f es la función objetivo. Sea $x \in \mathbb{R}^n$, denotamos en general la norma del espacio, que en este caso es $X = \mathbb{R}^n$, como $\|x\| = \|x\|_X$, aunque en general trabajaremos con la norma euclídea que de ser tal caso, lo denotaremos como $\|x\|_2$. Paralelamente, dado otro elemento del espacio y, denotamos el producto escalar como

$$x^T y = (x, y) = \langle x, y \rangle = \sum_{i=1}^{n} x_i y_i.$$

Sea $z \in \mathbb{R}^n$, trabajaremos con la bola abierta centrada en z y de radio R, $B(z, R)$, como con su bola cerrada $\bar{B}(z, R)$, que las definimos como

$$B(z, R) = \{ y \in \mathbb{R}^n \mid \|z - y\| < R \}$$
$$\bar{B}(z, R) = \{ y \in \mathbb{R}^n \mid \|z - y\| \leq R \}.$$

X^c denotará el complementario del conjunto X, que obviamente en el caso de $\mathbb{R}^n$ será el conjunto vacío $\emptyset$. $Int(X)$,$\overline{X}$ y δX corresponderán con el interior, la adherencia y la frontera, respectivamente de X, cuya definición es la siguiente:

$$int(X) = \{ x \in X \mid \exists R > 0, \text{ con } B(x, R) \subset X \}$$
$$\overline{X} = \{ x \in X \mid \forall R > 0, \ B(x, R) \cap X \neq \emptyset \}$$
$$\delta X = \overline{X} \cap \overline{X^c}.$$

Sea ahora $f : \mathbb{R}^n \to \mathbb{R}$ una función al menos C^2. Entonces definimos la "matriz Hessiana" de f, y lo escribiremos como $H(f), Hf, H_f$, como la matriz cuyas entradas son todas las derivadas parciales de f de segundo orden:

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \cdots & \ddots \end{bmatrix}$$

.

Definición 2.1. *Dada una función $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$, diremos que f es (globalmente) **Lipschitz** en el conjunto U cuando existe una constante $L > 0$ tal que*

$$\|f(x) - f(y)\| \leq L\|x - y\|, \ \forall x, y \in U$$

.

Proposición 2.2. ***(Inecuación de Jenssen)*** *Sea $f(x)$ una función convexa y sea $w_1, w_2, ..., w_n$ pesos tales que*

- $w_j \geq 0$.
- $\sum_{j=1}^{n} w_j = 1$.

Entonces, para cualesquiera $x_1, x_2, ..., x_n$ obtenemos la siguiente desigualdad

$$f(w_1x_1 + w_2x_2 + ... + w_nx_n) \leq w_1f(x_1) + w_2f(x_2) + ... + w_nf(x_n).$$

La prueba de esta inecuación la podemos encontrar en Andrew D. Smith, Convex Sets and Jensen?s Inequality. School of Mathematics and Statistics. University College Dublin.

Capítulo 3

Optimización convexa

En este capítulo abarcaremos las definiciones básicas de convexidad relativas a funciones y conjuntos, además de la importancia de la optimización convexa. Por otro lado, y para terminar, daremos paso a una serie de cuestiones en torno a la complejidad de un algoritmo convexo.

3.1. Convexidad: Nociones básicas y su importancia en optimización

Definición 3.1. *Sean x, y puntos de $\mathbb{R}^n$ distintos. Llamaremos* ***recta que pasa por*** *x* ***e*** *y al formado por los puntos de la forma $z = \theta y + (1-\theta)x$ con $\theta \in \mathbb{R}$. Cuando el parámetro $\theta \in [0,1]$ lo llamaremos la recta que empieza en x y termina en y, de manera que, cuando $\theta = 0$ tenemos $z = x$ y cuando $\theta = 1$ tenemos $z = y$.*

Definición 3.2. *Un conjunto $X \subseteq \mathbb{R}^n$ se dice que es* ***convexo****, si $\forall x, y \in X$, se tiene que la recta que pasa por x e y pertenece a X, es decir $\theta y + (1-\theta)x \in X$, $\forall \theta \in [0,1]$.*

Pero la noción de convexidad no sólo es exclusivo de conjuntos, también podemos relacionarlo con otros objetos matemáticos, como las funciones.

Definición 3.3. *Una función $f : X \to \mathbb{R}^n$ es* ***convexa*** *si el* ***dom f*** *es un conjunto convexo y $\forall x, y \in dom\ f$, $\forall \theta \in [0,1]$, tenemos que*

$$f(\theta x + (1+\theta)y) \leq \theta f(x) + (1+\theta) f(y)$$

.

Geométricamente, esta inecuación quiere decir que el segmento entre $(x, f(x))$ y $(y, f(y))$. La función f es estrictamente convexa, si la inecuación anterior se satisface con menor estricto, cualesquiera que sea $x \neq y$ y $\theta \in (0,1)$. Decimos que f es ***cóncava*** *si $-f$ es convexa, y* ***estrictamente cóncava*** *si $-f$ es estrictamente convexa.*

A menudo, es conveniente extender una función cóncava a todo $\mathbb{R}^n$ definiéndola en los puntos que no pertenecen al dom f como ∞. Si f es convexa, definiremos la extensión de f como $g : X \to \mathbb{R} \cup \infty$ tal que

$$g(x) = \begin{cases} f(x) & \text{si } x \in \text{dom } f \\ \infty & \text{si } x \notin \text{dom } f. \end{cases}$$

La extensión g, por tanto, está definida sobre todo $\mathbb{R}^n$ y toma valores en $\mathbb{R} \cup \infty$. De aquí en adelante, nuestro objetivo será minimizar cierta función real $f(x)$, definida sobre un subconjunto $X \subseteq \mathbb{R}^n$, con $x \in X$. Nos centraremos principalmente en minimizar ciertas funciones para las que no es posible obtener una solución analítica de $\nabla f(x) = 0$, de manera que, se recurre a un proceso iterativo por el que obtenemos puntos del dominio de f que se aproximan a dicho mínimo o cuya imagen se acerca a la imagen del mínimo. Hemos de tener en cuenta, que habrá ciertas funciones que no sean diferenciables, y por tanto, no podamos resolver dicha ecuación, aunque intentaremos de alguna manera, mediante un proceso iterativo, aproximarnos lo máximo posible al mínimo de dicha función. Sí es verdad, que el hecho de estudiar funciones convexas sobre conjuntos convexos $X \subseteq \mathbb{R}^n$ nos llevarán a algoritmos más rápidos, los cuales se llevarán todo el foco de nuestro atención. Para ello, vamos a estudiar algunos resultados sobre funciones convexas, que nos serán de gran utilidad.

Proposición 3.4. *(**Mínimo global.**) Sean $X \subseteq \mathbb{R}^n$ un conjunto convexo, $f : X \to \mathbb{R}$ una función y $x \in X$ un mínimo local de f. Si f es convexa, entonces x es un **mínimo global de f.***

Demostración. Sea f una función convexa definida sobre un conjunto convexo $X \subseteq \mathbb{R}^n$ y sea $x \in X$ un mínimo local de f. Dado $y \in X$, sea $\theta > 0$ de manera que $(1+\theta)x + \theta y$ pertenezca al entorno de x donde éste minimiza la función, entonces la convexidad de f implica que $f(x) \leq f((1+\theta)x + \theta y) \leq (1+\theta)f(x) + \theta f(y)$, y por tanto, $f(x) \leq f(y)$. □

Proposición 3.5. *(**Caracterización del mínimo de una función convexa.**) Dados $X \subseteq \mathbb{R}^n$ conjunto convexo y cerrado,sea X_1 un conjunto abierto de R^n con $X \subset X_1$, y $f : X \to \mathbb{R}$, tal que f es diferenciable en X_1. Entonces:*

$$x^* \in argmin\ _{x \in X} f(x) \Leftrightarrow f(x^*)^T(x^* - x) \leq 0, \quad \forall x \in X$$

Demostración.

$\boxed{\Rightarrow}$ Consideremos $x \in X$ y una función real $g_x(t) := f(x^* + t(x - x^*)) \leq 0$ cuya derivada es $\nabla g_x := \nabla f(x^* + t(y - x^*))^T(y - x^*)$.

Como $g_x(0) := f(x^*)$ es un punto mínimo de la función, va a existir un entorno del 0 en el que $\nabla g_x(t) := \nabla f(x^* + t(y - x^*))^T(y - x^*) \geq 0$, es decir, existe $\delta \geq 0$ tal que $\nabla g_x(t) := \nabla f(x^* + t(y - x^*))^T(y - x^*) \geq 0, \forall x$ tal que $|x| < \delta$. Es decir, que $g_0' = f(x^*)^T(y - x^*)' \geq 0$.

$\boxed{\Leftarrow}$ En primer lugar, para probar esta implicación, lo que haremos es probar que $f(x) - f(y) \leq f(x)^T(x - y) \forall x, y \in X$.

Sea $\phi \in [0, 1]$ y sea $h = \phi(y - x)$ $\forall x, y \in X$ basta con probar que se verifica que (por la convexidad de f)

$$\begin{aligned} f(y) &\geq f(x) + \frac{f((1-\phi)x + \phi y) - f(x)}{\phi} \\ &= f(x) + \frac{f(x+h) - f(x)}{h}(y - x) \\ &\to f(x) + \nabla f(x)^T(y - x) \text{ (si } \phi \to 0) \text{ (*)} \end{aligned}$$

Si existe $x^* \in X$ tal que $\forall y \in \mathbb{R}^n, \nabla f(x^*)^T(x^* - y) \leq 0$ aplicando (*)
$\Rightarrow f(x^*) \leq f(y)$

□

Proposición 3.6. *Sean $f_1, ..., f_m : \mathbb{R}^n \to \mathbb{R}$ funciones convexas y sean $t_1, ..., t_m$ números reales. Supongamos que existe $x_0 \in \mathbb{R}^n$ con $f_i(x_0) < \infty \ \forall i = 1, ..., m$.*
Entonces,

$$f(x) = \sum_{i=0}^{m} t_i f_i(x) \tag{3.1}$$

es una función convexa.

Demostración. La convexidad se obtiene directamente de la definición. Por ser f una función cerrada, se tiene que:

$$\liminf_{y \to x} \ t_j f_j(y) = t_j \liminf_{y \to x} \ f_j(y).$$

El resultado se tiene a raíz de que,

$$\sum_{j=1}^{m} \liminf_{y \to x} \ t_j f_j(y) = \liminf_{y \to x} \sum_{j=1}^{m} t_j f_j(y) \tag{3.2}$$

□

Definición 3.7. *El **grafo** de una función $f : \mathbb{R}^n \to \mathbb{R}$ está definido como:*

$$graf\ f = \{(x, f(x)) | x \in \textbf{\textit{dom f}}\},$$

*el cual, es un subconjunto de $\mathbb{R}^{n+1}$. El **epígrafo** de una función $f : \mathbb{R}^n \to \mathbb{R}$ está definido como*

$$epi\ f = \{(x, t) | x \in dom\ f, f(x) \leq t\},$$

el cual, es un subconjunto de $\mathbb{R}^{n+1}$.

Teorema 3.8. *Una función es convexa si y solo si su epígrafo es un conjunto convexo.*

Demostración.

$\boxed{\Rightarrow}$ Para la condición necesaria, basta tomar dos puntos $(x', t'), (x'', t'') \in$ epi f, con $\phi \in [0, 1]$, luego:

$$f((1 - \phi)x' + \phi x'' \leq (1 - \phi)f(x')) + \phi f(x'') \leq (1 - \phi)t' + \phi t''. \tag{3.3}$$

$\boxed{\Leftarrow}$ Para la condición suficiente, dados $x', x'' \in X$, entonces se tiene que:
$(x', f(x')), (x'', f(x'')) \in$ epi f, por ser éste un conjunto convexo.

□

3.2. Algunas notaciones sobre la convergencia

En el siguiente capítulo, abordaremos un método iterativo llamado **método de descenso por gradiente**, y otras variantes, el cual, tendrá como objetivo optimizar ciertas funciones f. Hasta aquí todo bien, pero sí es cierto, que en algunos casos no tendremos dicha función como dato, sino su imagen $f(x)$ o su gradiente $\nabla f(x)$, por tanto, es conveniente y de interés, estudiar la convergencia del **minimizante o minimizador** de la función que éstos generan. Para estudiar dicha convergencia, será un requisito ver el número de iteraciones necesarias para aproximarnos lo suficiente a nuestro objetivo, de manera que, sea lo suficientemente próximo para cierto $\varepsilon>0$. A tal punto de nuestro dominio que se adecúe a dicha distancia de ε lo llamaremos **ε-aproximante**. Por otro lado, el coste computacional quedará fijado por el número de iteraciones que nos llevará a evaluar, en cada paso iterativo, para así obtener nuestro minimizador. Se hará uso de la notación O de Landau indicando que un número de iteraciones p(ε) es $O(q(\varepsilon))$, y se llamará **orden de convergencia** al número de iteraciones del método que necesita $p(\varepsilon)$ para llegar a un ε-aproximante.

Capítulo 4

Métodos de descenso por gradiente y otras variantes

El algoritmo descrito en este capítulo produce una secuencia minimizante x_k, donde

$$x_{k+1} = x_k + t_k \Delta x_k \tag{4.1}$$

y $t_k > 0$. Aquí, la concatenación de los símbolos Δ y x que forman Δx ha de ser leído como una indentidad singular, es decir, un vector en $\mathbb{R}^n$ llamado *paso o dirección de búsqueda* (no es necesario que sea unitario). El escalar $t_k \geq 0$ es llamado *longitud de paso* en la iteración k. Cuando nos centremos en la iteración del algoritmo, a veces suprimimos el subíndice y usamos la notación $x^+ = x + t\Delta x$ en lugar de $x_{k+1} = x_k + t_k \Delta x_k$. Todos los métodos que estudiamos son métodos descendientes, lo que quiere decir que:

$$f(x_{k+1}) < f(x_k) \tag{4.2}$$

excepto cuando x_k sea el óptimo.

En gran parte de este capítulo, asumiremos que la función objetivo es fuertemente convexa en S (ver 2.1), lo que quiere decir que $\exists m > 0$ tal que

$$\nabla^2 f(x) \geq mI \tag{4.3}$$

para todo $x \in S$. La convexidad fuerte tiene numerosas consecuencias, por ejemplo, para $x, y \in S$, tenemos que

$$f(y) = f(x) + \nabla^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(z)(y - x) \tag{4.4}$$

para cierto z en el segmento que une x con y. Por asumir la convexidad fuerte (4.3), el último término del lado derecho es de un orden de al menos $(m/2)|y - x\|_2^2$, por lo que tenemos la inecuación siguiente

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2 \tag{4.5}$$

para todo x e y en S. Cuando $m = 0$, tenemos la inecuación característica de la convexidad; para $m > 0$ obtenemos un límite inferior en $f(y)$ mejor que el que sigue para la convexidad.

El término de la derecha de (4.5) es una función cuadrática para y (con x fijo). Imponiendo que el gradiente en y es igual a cero, obtenemos que $y^\star = x-(1/m)\nabla f(x)$ es el punto óptimo que minimiza el lado derecho, por lo tanto, tenemos que

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|_2^2 \\ &\geq f(x) + \nabla f(x)^T(y^\star - x) + \|y^\star - x\|_2^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 \end{aligned}$$

$\forall y \in S$. Luego,

$$p^\star = f(y^*) \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 \tag{4.6}$$

Esto último nos dice que si el gradiente en un punto es pequeño, entonces dicho punto está próximo al óptimo. La inecuación de (4.6) podemos verla como una condición de suboptimalidad que generaliza la condición de optimalidad $\nabla f(x) = 0$:

$$||\nabla f(x)||_2 \leq (2m\varepsilon)^{\frac{1}{2}} \Rightarrow f(x) - p^\star \leq \varepsilon \tag{4.7}$$

Por otro lado, la inecuación (4.5) implica que los subconjuntos contenidos en S están acotados, y por tanto, S también. Por lo tanto, el valor máximo de $\nabla^2 f(x)$, la cual es una función continua de x en S, está acotada superiormente en S, es decir, $\exists M > 0$ tal que

$$\nabla^2 f(x) \geq MI \tag{4.8}$$

$\forall x \in S$. Esta cota superior del Hessiano implica que para cualquier $x, y \in S$,

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{M}{2}\|y-x\|_2^2 \tag{4.9}$$

Minimizando cada lado de la inecuación sobre y, tenemos que

$$p^\star \geq f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2 \tag{4.10}$$

El esquema de un método de descenso general es el que sigue a continuación, el cual, alterna dos pasos: determinar la dirección de descenso Δx, y seleccionar la longitud de paso t.

Proposición 4.1. *(Algoritmo general de un método de descenso)*

1. ***Dar un punto inicial*** $x \in$ ***dom*** f .
2. ***Repetir***
   a) *Determinar la dirección de descenso* Δx
   b) *Línea de búsqueda. Elegir una longitud de paso* $t > 0$.
   c) *Update.* $x := x + t\Delta x$.
3. ***Parar cuando el criterio se satisfaga.***

Un método de descenso práctico tiene la misma estructura general, pero puede ser organizada de diferente manera. Por ejemplo, el criterio de parada es, frecuentemente, checkeado o inmediatamente después de que la dirección de descenso Δx es compilada. Éste, normalmente, es de la forma $\|\Delta f(x)\|_2 \leq \beta$, donde β es un número pequeño positivo, que suele ser sugerido por la condición de suboptimalidad.

En particular, abarcaremos los métodos iterativos de descenso por gradiente (DG) y descenso por gradiente proyectado (DGP), para resolver el problema de optimización de

$$\text{mín } f(x) \tag{4.11}$$

donde $f : \mathbb{R}^n \to \mathbb{R}$ es convexa y C^2 (dos veces difrenciable, lo que implica que el dom f sea abierto). Vamos a asumir que el problema es resoluble, por tanto, $\exists x_* \in$ dom f punto óptimo. Posteriormente, en uno de los capítulos siguientes, veremos que esto implicará que dicho x_* existe y es único. Además, como f es diferenciable y convexa, una condición necesaria y suficiente para x_* será que $\nabla f(x_*) = 0$.

Como su nombre indica, el (DG) y (DGP) es un método de descenso que requiere del punto o iterante anterior de dicho proceso, es decir,

$$x_{k+1} = x_k + \alpha_k \Delta x_k \tag{4.12}$$

aunque bien es cierto que en ciertos casos, resolver el problema (4.11) es equivalente a resolver analíticamente el problema $\nabla f(x^*) = 0$, pero nosotros lo haremos mediante un algoritmo iterativo. Lo que quiere decir que, nuestro algoritmo computará una serie de puntos $x_0, x_1, x_2, ... \in$ dom f, con $\lim_{k\to\infty} f(x_k) = p_*$. Dicha sucesión de puntos la llamaremos *secuencia minimizante*, de manera que, el algoritmo habrá acabado cuando $f(x_k) - p_* \leq \varepsilon$, para cierto ε>0 requerido.

La dirección del (DG) podemos intuir que tiene como objetivo un punto del dominio de f, pero no siempre es así, en el caso del (DGP) dicha dirección nos conducirá a un punto x_e que no pertenece al dominio. A partir de estos, se desarrollarám más métodos, que nos permitirán solucionar otros problemas que no podamos con alguno de los ya mencionados, como *método de descenso por gradiente acelerado de Nesterov o el método de Frank-Wolfe.*

4.1. Descenso por gradiente y descenso por gradiente proyectado

El ***método de descenso por gradiente*** es uno de los métodos más empleados y conocidos para hallar el mínimo de cierta función $f : \mathbb{R}^n \to \mathbb{R}$ diferenciable. Una elección natural (y como su nombre bien indica) es utilizar el gradiente negativo de x, es decir, $\Delta x = -\nabla f(x)$. El algoritmo parte de cierto punto $x_0 \in \text{dom } f$, de manera que, $\forall k \in \mathbb{N} \cup 0$:

1. **Dirección.** $\Delta x_k = -\nabla f(x_k)$
2. **Test de parada.** Elija el tamaño de paso ε>0 a través de la búsqueda de línea exacta o retroactiva
3. **Update.** $x_{k+1} = x_k + \alpha_k \Delta x_k$
4. Parar cuando el test de parada se satisfaga.

El criterio que usualmente usaremos, de parada, será de la forma $\|\nabla f(x_k)\|_2 \leq \beta$, donde β es positivo y pequeño. En la mayoría de los casos, esta condición se chequea previo al paso de Update. Por otro lado, la razón de usar la dirección del método como $\nabla f(x_k)$ es porque es la dirección de máximo descenso, lo cual, se puede comprobar fácilmente.

En lo referente a la convergencia del método, usaremos la notación $x^+ = x + t\Delta x$ en vez de $x_{k+1} = x_k + t_k \Delta x_k$ donde, como ya dijimos antes $\Delta x_k = -\nabla f(x_k)$. Vamos a asumir que f es una función fuertemente convexa en S, por tanto, existen unas constantes positivas m y M tales que $mI \leq \nabla^2 f(x) \leq MI$, $\forall x \in S$. Además definimos la función $f^\star : \mathbb{R} \to \mathbb{R}$ como $f^\star = f(x - t\nabla f(x))$, donde $f^\star$ es una función con longitud de paso t en la dirección negativa del gradiente. De aquí en adelante, consideraremos exclusivamente t para los cuales, $x - t\nabla f(x) \in S$. Entonces, por la inecuación

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{M}{2}\|y - x\|_2^2 \tag{4.13}$$

que es análogo a

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2. \tag{4.14}$$

Con $y = x - t\nabla f(x)$, obtenemos una cota cuadrática superior para $f^\star$:

$$f^\star \leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2. \tag{4.15}$$

Ahora asumimos que se usa una línea de búsqueda exacta, y minimizamos en t en ambos lados de la inecuación (4.15). En el lado izquierdo obtenemos $f^\star(t_{exact})$ donde t_{exact} es la longitud de paso que minimiza $f^\star$. En el lado derecho tenemos un cuadrado, el cual, se minimiza tomando $t = \frac{1}{M}$, que tiene como valor mínimo $f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2$.

Por lo tanto, tenemos que

$$f(x^+) = f^\star(t_{exact}) \leq f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2.$$

Que restando $p^\star$ a ambos lados, obtenemos

$$f(x^+) - p^\star \leq f(x) - p^\star - \frac{1}{2M}\|\nabla f(x)\|_2^2. \tag{4.16}$$

Teniendo en cuenta esto, junto a que $\|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^\star$ podemos concluir lo siguiente

$$f(x^+) - p^\star \leq (1 - \frac{m}{M})(f(x) - p^\star)$$

Aplicando la inecuación recursivamente, nos percatamos de que

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star) \tag{4.17}$$

donde $c = 1 - m/M < 1$, el cual nos dice que $f(x^{(k)})$ converge a $p^\star$ cuando $k \to \infty$. En particular, debemos tener que $f(x^{(k)}) - p^\star < \varepsilon$ después como máximo

$$\frac{log((f(x^{(0)}) - p^\star)/\varepsilon)}{log(1/c)} \tag{4.18}$$

iteraciones del método del gradiente con linea de búsqueda exacta. Esta cota del número de iteraciones requerido, puede darnos algo de información sobre el método de descenso de gradiente. El numerador,

$$log((f(x^{(0)})) - p^\star)/\varepsilon)$$

puede ser interpretado como el *log* del radio de la suboptimalidad inicial hasta la final. Esto sugiere que el número de iteraciones depende de cuánto de bueno es el punto inicial, y cuál es la precisión final requerida.

El denominador que aparece en (4.18), $log(1/c)$, es una función de M/m, la cual puede ser vista como un límite en el número de condición de $\nabla^2 f(x)$ sobre S, o el número de condición de los conjuntos de subnivel $z|f(x) \leq \alpha$. Además, tenemos que

$$log(1/c) = -log(1 - m/M) \approx m/M,$$

por tanto, nuestro límite del número de interaciones requeridas aumenta aproximadamente de forma lineal conforme aumenta M/m. Más adelante veremos que el método del gradiente, de hecho requiere un gran número de iteraciones cuando el Hessiano de f, próximo a $x^\star$, tiene un gran número de condiciones. En cambio, cuando los conjuntos de subnivel de f son relativamente isotrópicos, entonces el número de condición del límite de M/m puede ser elegido para que sea relativamente pequeño, de manera que, en (4.18) muestra que la convergencia es rápida, tanto como c sea pequeño, o al menos no demasiado cercano a uno. Este límite muestra que el error $f(x^{(k)}) - p^\star$ converge a cero al menos, tan rápido como la serie geométrica. En el contexto de métodos numéricos iterativos, es llamado usualmente como *convergencia lineal*, dado que el error se encuentra debajo de una línea en una gráfica log-lineal de error versus número de iteración.

Ejemplo 4.2. *Nuestro primer ejemplo va a ser muy sencillo. Consiste en un problema en $\mathbb{R}^2$ de una función objetivo cuadrática:*

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

donde $\gamma > 0$. Se ve fácilmente que el punto óptimo es $x^\star = 0$, cuyo valor es 0. Se tiene además que, el Hessiano de f es constante, cuyos autovalores son 1 y γ, por lo que, el número de condiciones de los conjuntos de subnivel de f son todos exactamente

$$\frac{\text{máx}(1,\gamma)}{\text{mín}(1,\gamma)} = \text{máx}\left(\gamma, \frac{1}{\gamma}\right)$$

Las elecciones más estrictas para la constantes de la convexidad fuerte m y M son

$$m = \text{mín}\Big(1,\gamma\Big), M = \text{máx}\Big(1,\gamma\Big).$$

Aplicamos el método de descenso por gradiente con línea de búsqueda exacta, empezando en el punto $x^{(0)} = (\gamma, 1)$. En este caso, podemos derivar las siguientes expresiones de forma cerrada para las iteraciones $x^{(k)}$ y sus valores correspondientes:

$$x_1^{(k)} = \gamma\Big(\frac{\gamma-1}{\gamma+1}\Big)^k, x_2^{(k)} = \Big(\frac{1-\gamma}{\gamma+1}\Big)^k,$$

y

$$f(x^{(k)}) = \frac{\gamma(\gamma+1)}{2}\Big(\frac{\gamma-1}{\gamma+1}\Big)^{2k} = \Big(\frac{\gamma-1}{\gamma+1}\Big)^{2k} f(x^{(0)})$$

Para este ejemplo, la convergencia es exactamente lineal, es decir, el error es exactamente una serie geométrica, reducida por el factor $|(\gamma-1)/(\gamma+1)|^2$ en cada iteración. Para $\gamma = 1$, la solución exacta se encuentra en la primera iteración; para γ distinto de 1 la convergencia es rápida. La convergencia puede llegar a ser muy lenta para $\gamma \ll 1$. Sin embargo, en la mayoría de los problemas de optimización, la función f está definida en $X \subsetneq \mathbb{R}^n$, por lo que tenemos la certeza de que $x^{k+1} \in X$. La manera más obvia cuando el punto no esté en X, es la de tomar el punto más próximo que pertenezca a X. Sí es verdad que no se puede garantizar la unicidad de dicho punto, a no ser que X cumpla ciertas propiedades, en la que vuelve a aparecer la convexidad.

Teorema 4.3. *Si H es un espacio de Hilbert, y si $C \subset H$ es un subconjunto convexo y cerrado, entonces existe un único $\overline{x} \in C$ tal que $\|\overline{x}\| \leq \|y\|$, $\forall y \in C$.*

Demostración. Sea $\delta = \text{ínf}\{\|x\| : x \in C\}$. Hemos de demostrar que existe un único vector $x \in C$ tal que $\|x\| = \delta$. Vamos a probar en primer lugar la unicidad. Sean $x, y \in C$. Aplicando la identidad del paralelogramo a $x/2$ e $y/2$, obtenemos

$$\frac{1}{4}\|x-y\|^2 = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2 - \|\frac{x+2}{2}\|^2.$$

Como $\frac{x+y}{2} \in C$, tenemos

$$\|x-y\|^2 \leq 2\|x\|^2 + 2\|y\|^2 - 4\delta^2. \ (*)$$

Si ambos vectores x e y son de norma mínima, se tendría $\|x\| = \|Y\| = \delta$, y deduciríamos de (*) que $\|x-y\| = 0$, así que $x = y$. Esto prueba la unicidad.

En cuanto a la existencia, tomemos una sucesión $(x_n) \subset C$ tal que $\|x_n\| \to \delta$. Fijemos $\varepsilon > 0$ y seleccionemos un número $\eta \in (0, \text{mín}\{\varepsilon/4, \varepsilon^2/(16\delta)\})$. Podemos elegir un n_0 tal que $\|x_n\| < \delta + \eta$ para todo $n \geq n_0$. De nuevo aplicamos (*), pero esta vez a los vectores x_m, x_n con $m, n \geq n_0$. Resulta

$$\|x_m - x_n\|^2 \leq 2\|x_m\|^2 + 2\|x_n\|^2 - 4\delta^2 < 4(\delta+\eta)^2 - 4\delta^2 = 4\eta^2 + 8\delta\eta < \frac{\varepsilon^2}{4} + \frac{\varepsilon^2}{2} < \varepsilon^2.$$

Esto implica que (x_n) es de Cauchy. Por ser H completo, existe un $x_0 \in H$ tal que $x_n \to x_0$. Debido a que C es cerrado, $x_0 \in C$. Por último, gracias a la continuidad de la norma, $\|x_n\| \to \|x_0\|$, luego, por la unicidad del límite, $\|x_0\| = \delta$. □

Teorema 4.4. ***(Teorema de la proyección convexa)*** *Sea $X \subset \mathbb{R}^n$ un conjunto convexo, no vacío y cerrado y sea $y \in \mathbb{R}^n$. Entonces*

$$\exists!\ \overline{x} \in X \text{ tal que } \|y - \overline{x}\| \leq \|y - x\|,\ \forall x \in X.$$

Demostración. Usando el teorema mencionado, basta definir el conjunto $C := \{y\} - K$. En efecto, sea $\overline{y} \in C$ tal que $\|\overline{y}\| \leq \|z\|$,$\forall z \in C$, sea $\overline{x} := y - \overline{y}$, sea $x \in K$ y $z := y - x \in C$. Está claro entonces que $\|y - \overline{x}\| = \|\overline{y}\| \leq \|z\| = \|y - x\|$. □

El teorema de la proyección convexa adquiere un papel fundamental, pues va a ser la llave para adaptar el método de descenso por gradiente a numerosos problemas de optimización con restricciones, dando lugar al siguiente método a desarrollar, el *método de descenso por gradiente conjugado*. Antes de dar paso con esto, vamos a introducir una serie de resultados necesarios que son de gran relevancia para entender la geometría de los conjuntos convexos, entre otras cosas.

Dado un conjunto convexo cerrado C, definimos la aplicación $\Pi_C : \mathbb{R}^n \to C$ como la que asocia a cada punto $x \in \mathbb{R}^n$ el único punto de C que está a mínima distancia de x. A la aplicación Π_C la denominamos como **la proyección de un punto sobre un conjunto**.

Definición 4.5. *Sean $X \subset \mathbb{R}^n$ y $f\colon X \to \mathbb{R}^n$. Se denomina **subgradiente de f en x** a todo $\xi \in \mathbb{R}^n$ tal que para todo $y \in \mathbb{R}^n$*

$$f(x) - f(y) \leq \xi^T(x - y)$$

El conjunto de subgradientes de f en x se denomina ***subdiferencial de f en x*** *y se denota por* $\partial f(x)$

$$\partial f(x) = \{\xi \in \mathbb{R}^n : f(x) - f(y) \leq \xi^T(x-y)\}$$

Teorema 4.6. ***(Teorema del hiperplano separante)*** *Sea* $X \subset \mathbb{R}^n$ *un conjunto convexo, no vacío y cerrado. Sea* $y \notin X$*. Entonces existe* $p \in \mathbb{R}^n, p \neq 0$ *y* $\alpha \in \mathbb{R}$ *tales que*

- $p^T x \leq \alpha$*, para todo* $x \in X$*.*
- $p^T y > \alpha$*.*

Demostración. Sea $\overline{x}$ la proyección de y sobre X. Entonces $(y-\overline{x})^T(x-\overline{x}) \leq 0, \forall x \in X$. Sea $p = y - \overline{x}$ y $\alpha = p^T\overline{x}$. Veamos que se cumplen las condiciones requeridas:

- Si $x \in X$, $p^T x = (y-\overline{x})^T(x-\overline{x}) + \alpha \leq \alpha$.
- $p^T y = (y-\overline{x})^T(y-\overline{x}) + \alpha = \|y-\overline{x}\|^2 + \alpha > \alpha$.

□

Teorema 4.7. ***(Teorema del hiperplano soporte)*** *Sea* $X \subset \mathbb{R}^n$ *un conjunto convexo. Sea* $x^* \in \partial X$*, frontera de* X*. Entonces existe* $p \in \mathbb{R}^n, p \neq 0$*, tal que*

$$p^T(x - x^*) \leq 0, \ \forall x \in X.$$

Demostración. Se sabe que $x^* \notin \text{int}(X) = \text{int}(\overline{X})$. Además, se tiene que para todo $k \in \mathbb{N}$, $\exists y_k \in B(x^*, \frac{1}{k}) \cap \overline{X}^c$. Existe $p_k \in \mathbb{R}^n$, $\|p_k\| = 1$, con $p_k^T y_k > p_k^T x$, para todo $x \in \overline{S}$ * . Tenemos que $y_k \to x^*$. Para una subsucesión, $p_k \to p$ con $\|p\| = 1$. Tomando límites en *, $p^T(x - x^*) \leq 0$, para todo $x \in \overline{X}$. □

Proposición 4.8. *Sea* $X \subset \mathbb{R}^n$ *convexo, abierto y no vacío, y sea* $f : X \to \mathbb{R}$ *diferenciable y convexa en un punto* $x_0 \in$ *int(X). Entonces* $\partial f(x_0) = \{\nabla f(x_0)\}$*.*

Demostración. Como f es convexa, entonces $\partial f(x_0) \neq \emptyset$ (ya que se tiene que si una función f es convexa, entonces el subdiferencial de f, en este caso, en x_0, es no vacío). Sea $\beta \in \partial f(x_0)$, $d \in \mathbb{R}^n$ y $\lambda > 0$. Por definición de subgradiente

$$\frac{f(x_0 + \lambda d) - f(x_0)}{\lambda} \geq \beta^T d$$

Haciendo $\lambda \to 0$, dado que f es diferenciable, se obtiene que $\nabla f(x_0) \geq \beta^T d$ para toda dirección $d \in \mathbb{R}^n$. Sólo cabe que $\partial f(x_0) = \nabla f(x_0)$. □

Con esta última proposición que acabamos de ver, hemos obtenido que cuando la función es diferenciable, el subdiferencial es un conjunto formado por un único punto, que no es más que el propio gradiente en dicho punto, $\nabla f(x)$. Este último resultado, juntos a los otros anteriores, permite adaptar el descenso por gradiente a nuevos tipos de situaciones, concluyendo así, con un nuevo método, el de *descenso por gradiente proyectado* que solventa las nuevas situaciones planteadas.

Pasamos ahora por lo tanto, a describir el método de descenso por gradiente proyectado.Vamos a suponer que el conjunto $X \subset \mathbb{R}^n$ está contenido en la bola

euclídea cuyo centro es el punto inicial x_0 y de radio $r > 0$. Además, asumiremos que f es tal que para cualquier $x \in X$, y para cualquier $g \in \partial f(x)$ (suponemos que $\partial f(x) \neq 0$), se tiene que $\|g\| \leq L$. Esta última suposición implica que la función f es L-lipschitziana. En este contexto, haremos dos modificaciones con respecto al método de descenso por gradiente. En primer lugar, reemplazaremos $\Delta f(x)$ (que puede no existir) por un subgradiente $g \in \partial f(x)$. Por otro lado, nos aseguraremos que el punto levantado esté en X proyectándolo de nuevo sobre él. Partimos entonces del punto inicial x_0, el método continúa de la siguiente manera ($n \geq 0$):

$$y_{n+1} = x_n - \mu g_n, g_n \in \partial f(x_n) \tag{4.19}$$
$$x_{n+1} = \Pi_X(y_{n+1}). \tag{4.20}$$

Lema 4.9. *Sea $x \in X$ e $y \in \mathbb{R}^n$, entonces*

$$(\Pi_X(y) - x)^T(\Pi_X(y) - y) \leq 0,$$

lo cual implica también que $\|\Pi_X(y) - x\|^2 + \|y - \Pi_X(y)\|^2 \leq \|y - x\|^2$.

La demostración de este lema la hemos tomado de Y.Nesterov, *Gradient methods for minimizing composite objetive function.* Core discussion papers, Université catholique de Louvain, Center for Operations Research And Econometrics (CORE),2007.

Teorema 4.10. *El método del gradiente proyectado con $\mu = \frac{r}{L\sqrt{n}}$ satisface que*

$$f\left(\frac{1}{n}\sum_{j=1}^{n} x_j\right) - f(x^*) \leq \frac{rL}{\sqrt{n}}.$$

Demostración. Usando la definición de subgradiente, la definición del método y la identidad elemental $2a^Tb = \|a\|^2 + \|b\|^2 - \|a - b\|^2$, obtenemos

$$\begin{aligned} f(x_j) - f(x^*) &\leq g_j^T(x_j - x^*) \\ &= \frac{1}{\mu}(x_j - y_{j+1})^T(x_j - x^*) \\ &= \frac{1}{2\mu}(\|x_j - x^*\|^2 + \|x_j - y_{j+1}\|^2 - \|y_{j+1} - x^*\|^2 \\ &= \frac{1}{2\mu}(\|x_j - x^*\|^2 - \|y_{j+1} - x^*\|^2) + \frac{\mu}{2}\|g_j\|^2. \end{aligned}$$

como $\|g_j\| \leq L$ y, por el lema 4.9

$$\|y_{j+1} - x^*\| \geq \|x_{j+1} - x^*\|.$$

Añadiendo la desigualdad resultante sobre j, y usando que $\|x_0 - x^*\| \leq r$

$$\sum_{j=1}^{n}(f(x_j) - f(x^*) \leq \frac{r^2}{2\mu} + \frac{\mu L^2 n}{2}.$$

Pegando el valor de μ directamente nos da directamente el resultado. □

Por tanto, observamos que el tamaño de paso recomendado en el teorema 4.10 depende del número de iteraciones a realizar. En la práctica, esto puede ser una característica indeseable. Por lo tanto, usando un tamaño de paso variable de la forma $\mu_s = \frac{r}{L\sqrt{s}}$ se puede probar la misma tasa hasta un factor $\lg t$. En cualquier caso estos tamaños de paso son muy pequeños, lo cual es una razón para que la convergencia sea lenta. En el siguiente apartado, veremos que asumiendo que la función f sea β-suave puede permitirnos que esta convergencia sea mucho más rápida. De hecho, a medida que uno se acerca, el tamaño óptimo de los propios gradientes tenderá a 0, lo que dará como resultado una especie de autoajuste de los tamaños de paso que no ocurre para una función arbitraria convexa.

Una función cuadrática es una función de la forma

$$J : v \in \mathbb{R}^n \to J(v) \in \mathbb{R}$$

donde $J(v) = \frac{1}{2}(Av, v) - (b, v)$ siendo A una matriz de n filas por n columnas simétrica y donde $b \in \mathbb{R}^n$. Una función cuadrática $J(.)$ es elíptica si y sólo si la matriz asociada A es definida positiva. Si A es definida positiva el mínimo de $J(.)$ está caracterizado por $\delta J(u) = 0$ es decir, $Au = b$. De modo que el problema de hallar $u \in R^n$ tal que $J(u) = \text{ínf}_{v \in \mathbb{R}^n} J(v)$ equivale a resolver $Au = b$. Vamos a considerar ahora la minimización de funciones cuadráticas con una matriz asociada A definida positiva y por tanto, elípticas. Sabemos que el problema tiene solución única. Una de las propiedades que tiene cualquier método de descenso es la siguiente:

- Cualquiera que sea la dirección de descenso d^n elegida, para ρ_n óptimo se tiene que para todo $n \geq 0$
$$(d^n, r^{n+1}) = 0$$
es decir, la dirección de descenso y el nuevo gradiente de la función son ortogonales.

- El problema de minimizar una función $J : v \to \frac{1}{2}(Av, v) - (b, v)$ con A simétrica y definida positiva, es quivalente a minimizar
$$E(v) = (A(v - u), v - u) = \|v - u\|_A^2$$

Nota 4.11. *Hemos introducido la notación $\|v\|_A = (Av, v)^{1/2}$ para la norma asociada al producto escalar $u, v \to (Au, v)$.*

Nota 4.12. *$E(\cdot)$ es la función error asociada al valor v, más precisamente, es el cuadrado de la norma asociada a la matriz A del error $e = v - u$.*

Tomaremos ahora $\Delta x_k = d^n$ y $\alpha_k = r^n$. Vamos a buscar ahora la nueva dirección de descenso d^n en el plano formado por las direcciones ortogonales r^n y d^{n-1}. Pongamos

$$d^n = r^n + \beta_n d^{n-1}$$

y calculemos el parámetro β_n de modo que el factor de reducción del error sea lo más grande posible. Tenemos

$$E(u^{n+1}) = E(u^n)\left(1 - \frac{(r^n, d^n)^2}{(Ad^n, d^n)(A^{-1}r^n, r^n)}\right)$$

Elegiremos β^n de manera que

$$\frac{(r^n, d^n)^2}{(Ad^n, d^n)(A^{-1}r^n, r^n)}$$

sea máximo. Como

$$(r^n, d^n) = (r^n, r^n + \beta_n d^{n-1}) = \|r^n\|^2 + \beta_n(r^n, d^{n-1}) = \|r^n\|^2$$

elegiremos $d^0 = r^0$ de modo que esta relación se verifique también para $n = 0$. Tendremos pues $(r^n, d^n) = \|r^n\|^2$ para todo $n \geq 0$. La determinación del máximo de

$$\frac{(r^n, d^n)^2}{(Ad^n, d^n)(A{-}1r^n, r^n)} = \frac{\|r^n\|^2}{(Ad^n, d^n)(A{-}1r^n, r^n)}$$

se reduce a minimizar (Ad^n, d^n). Desarrollamos este término

$$(Ad^n, d^n) = (A(r^n{+}\beta_n d^{n-1}), r^n{+}\beta_n d^{n-1}) = \beta_n^2(Ad^{n-1}, dn-1){+}2\beta_n(Ad^{n-1}, r^n){+}(Ar^n, r^n)$$

Para que esto último sea mínimo, hay que elegir β_n de modo que

$$\beta_n(Ad^{n-1}, d^{n-1}) + (Ad^{n-1}, r^n) = 0$$

de donde deducimos

$$\beta_n = -\frac{(Ad^{n-1}, r^n)}{Ad^{n-1}, d^{n-1})}$$

y también

$$(Ad^{n-1}, r^n + \beta_n d^{n-1}) = 0$$

es decir,

$$(Ad^{n-1}, d^n) = 0$$

4.1.1. Funciones β -suaves y α -fuertemente convexas

Diremos que una función f diferenciable es β**-suave** si el gradiente ∇f es β-Lipschitziano, es decir

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|.$$

Notemos que si f es dos veces diferenciable, entonces es equivalente a que los valores propios del Hessiano son más pequeños que β. En esta sección exploraremos mejoras potenciales en el radio de la convergencia bajo tales supuestos. Con el fin de evitar tecnicismos, consideraremos situaciones sin restricciones, donde f es una función convexa y β-suave en $\mathbb{R}^n$. El siguiente teorema muestra que el *método del gradiente*, cuya iteración es

$$x_{k+1} = x_k - \alpha_k \Delta f(x_k),$$

alcanza una velocidad de convergencia mucho más rápida que en el caso de funciones que no son β-suaves.

Teorema 4.13. *Sea f una función convexa y β-suave en $\mathbb{R}^n$. Entonces el método de descenso del gradiente con $\mu = \frac{1}{\beta}$ satisface que*

$$f(x_k) - f(x^*) \leq \frac{2\beta\|x_1 - x^*\|^2}{k-1}.$$

Previamente a la demostración del teorema, expondremos una series de propiedades de las funciones convexas β-suaves.

Lema 4.14. *Sea f una función β-suave en $\mathbb{R}^n$. Dados $x, y \in \mathbb{R}^n$,se tiene que*

$$|f(x) - f(y) - \nabla f(y)^T(x-y)| \leq \frac{\beta}{2}\|x-y\|^2.$$

Demostración. Representaremos $f(x) - f(y)$ como una integral, aplicando Cauchy-Schwarz y la β-suavidad:

$$\begin{aligned}
|f(x) - f(y) - \nabla f(y)^T(x-y)| &= \left|\left(\int_0^1 \nabla f(y + t(x-y))^T(x-y)\,dt\right) - f(y)^T(x-y)\right| \\
&\leq \left(\int_0^1 \|\nabla f(y + t(x-y)) - \nabla f(y)\| \cdot \|x-y\|\,dt\right) \\
&\leq \left(\int_0^1 \beta t\|x-y\|^2\,dt\right) \\
&= \frac{\beta}{2}\|x-y\|^2.
\end{aligned}$$

□

En particular, este lema muestra que si f es convexa y β-suave, para cualquier $x, y \in \mathbb{R}^n$, uno tiene que

$$0 \leq f(x) - f(y) - \nabla f(y)^T(x-y) \leq \frac{\beta}{2}\|x-y\|^2. \tag{4.21}$$

Esto nos da en particular la siguiente inecuación para evaluar la mejora en un paso del método de descenso del gradiente:

$$f(x - \frac{1}{\beta}\nabla f(x)) - f(x) \leq -\frac{1}{2\beta}\|\nabla f(x)\|^2. \tag{4.22}$$

El lema siguiente, el cual mejora la inecuación básica para el subgradiente bajo la suposición de β-suavidad, muestra el hecho de que f es convexa y β-suave si y sólo si (4.21) es cierto.

Lema 4.15. *Sea f tal que (4.21) es cierto. Entonces, para cualquier $x, y \in \mathbb{R}^n$, se tiene que*

$$f(x) - f(y) \leq \nabla f(x)^T(x-y) - \frac{1}{2\beta}\|\nabla f(x) - \nabla f(y)\|^2.$$

Demostración. Sea $z = y - \frac{1}{\beta}(\nabla f(y) - \nabla f(x))$. Entonces se tiene que

$$\begin{aligned} f(x) - f(y) &= f(x) - f(z) + f(z) - f(y) \\ &\leq \nabla f(x)^T(x - z) + \nabla f(y)^T(z - y) + \frac{\beta}{2}\|z - y\|^2 \\ &= \nabla f(x)^T(x - y) + (\nabla f(x) - \nabla f(y))^T(y - z) + \frac{1}{2\beta}\|\nabla f(x) - \nabla f(y)\|^2 \\ &= \nabla f(x)^T(x - y) - \frac{1}{2\beta}\|\nabla f(x) - \nabla f(y)\|^2. \end{aligned}$$

□

Podemos ahora probar el teorema 4.13

Demostración. Usando (4.22) y la definición del método se tiene que

$$f(x_{j+1} - f(x_j) \leq -\frac{1}{2\beta}\|\nabla f(x_j)\|^2.$$

En particular, denotando $\delta_j = f(x_j) - f(x^*)$, se tiene que:

$$\delta_{j+1} \leq \delta_j - \frac{1}{2\beta}\|\nabla f(x_k)\|^2.$$

Además, también se tiene que, por la convexidad

$$\delta_j \leq \nabla f(x_j)^T(x_j - x^*) \leq \|x_j - x^*\|\Delta\|\nabla f(x_j)\|.$$

Probaremos que $\|x_j - x^*\|$ decrece con j, lo que con las dos desigualdades anteriores implicará que

$$\delta_{j+1} \leq \delta_j - \frac{1}{2\beta}\|x_1 - x^*\|^2\delta_j^2.$$

Veamos cómo usar esta última desigualdad para concluir la prueba. Sea $\zeta = \frac{1}{2}\beta\|x_1 - x^*\|^2$, entonces

$$\zeta\delta_j^2 + \delta_{j+1} \leq \delta_j \Leftrightarrow \zeta\frac{\delta_j}{\delta_{j+1}} + \frac{1}{\delta_j} \Rightarrow \frac{1}{\delta_{j+1}} - \frac{1}{\delta_j} \geq \zeta \Rightarrow \frac{1}{\delta_t} \geq \zeta(t-1).$$

Por lo que sólo queda demostrar que $\|x_j - x^*\|$ decrece con j. Usando el Lema 4.14 inmediatamente se tiene que

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \frac{1}{\beta}\|\nabla f(x) - \nabla f(y)\|^2. \tag{4.23}$$

Lo usamos de la siguiente manera (junto con que $\nabla f(x^*) = 0$)

$$\begin{aligned} \|x_{j+1} - x^*\|^2 &= \|x_j - \frac{1}{\beta}\nabla f(x_j) - x^*\|^2 \\ &= \|x_j - x^*\|^2 - \frac{2}{\beta}\nabla f(x_j)^T(x_j - x^*) + \frac{1}{\beta^2}\|\nabla f(x_j)\|^2 \\ &\leq \|x_j - x^*\|^2 - \frac{1}{\beta^2}\|\nabla f(x_j)\|^2 \\ &\leq \|x_j - x^*\|^2, \end{aligned}$$

lo que concluye la prueba. □

Pasaremos ahora a hablar sobre otra propiedad de las funciones convexas que pueden significativamente acelerar la convergencia del primer orden de los métodos: **fuerte convexidad**. Diremos que una función $f : X \to \mathbb{R}$ es **α-fuertemente convexa** si ésta satisface la siguiente inecuación del subgradiente mejorado:

$$f(x) - f(y) \leq \nabla f(x)^T (x - y) - \frac{\alpha}{2}\|x - y\|^2. \tag{4.24}$$

Es claro que esta definición no requiere diferenciabilidad de la función f, y se puede reemplazar $\nabla f(x)$ en la inecuación de arriba por $g \in \partial f(x)$. Es inmediato comprobar que una función f es α-fuertemente convexa, si y solo si, $x \to f(x) - \frac{\alpha}{2}\|x\|^2$ es convexa (en particular, si f es dos veces diferenciable entonces los valores propios del Hessiano de f tienen que ser más grandes que α). La convexidad fuerte del parámetro α es una medida de la curvatura de f. Por ejemplo, una función lineal que no tiene curvatura y por tanto $\alpha = 0$. Por otro lado, se puede ver por qué a un valor grande de α conduciría una velocidad más rapida (de convergencia): en este caso, un punto alejado del óptimo hará que tenga un gradiente mayor y, como consecuencia, el método de descenso del gradiente dará grandes pasos cuanto más lejos se encuentre de éste. Si la función no es β-suave, habrá que tener cuidado y ajustar los tamaños de paso para que sean relativamente pequeños, pero no obstante, podremos mejorar la complejidad de la predicción desde $O(1/\varepsilon^2)$ a $O(1/\varepsilon\alpha)$.

Antes de ir a las pruebas, veamos otras de las interpretaciones de la convexidad fuerte y su relación con las funciones β-suaves. La ecuación (4.24) puede ser leída de la siguiente forma: para cualquier punto x, podemos encontrar una cota inferior $q_x^- = f(y) + \nabla f(y)^T (x - y) + \frac{\beta}{2}\|x - y\|^2$ para la función f, por ejemplo, $q_y^+ \geq f(x)$, $\forall x \in X$. Por tanto, en algún sentido, la convexidad fuerte es el *dual* de la suposición de la suavidad, y de hecho, esto se puede precisar en el marco de la *dualidad de Frenchel*. Además, es claro que siempre se tiene que $\beta \geq \alpha$.

Consideremos ahora el algoritmo del método del gradiente proyectado con tamaño de paso variable en el tiempo $(\mu_k)_{k\geq 1}$,

$$\begin{aligned} y_{k+1} &= x_k - \mu_k g_k,\ g_k \in \partial f(x_k) \\ x_{k+1} &= \Pi_X(y_{k+1}). \end{aligned}$$

Teorema 4.16. *Sea f α-fuertemente convexa y L-Lipschitziana en X. Entonces el método de descenso del gradiente proyectado con $\mu_k = \frac{2}{\alpha(s+1)}$ satisface*

$$f\left(\sum_{s=1}^{k} \frac{2s}{k(k+1)} x_s\right) - f(x^*) \leq \frac{2L^2}{\alpha(k+1)}. \tag{4.25}$$

Demostración. Mirando atrás en nuestro original análisis del método de descenso del gradiente proyectado y usando la hipótesis de fuerte convexidad, uno inmediatamente obtiene que

$$f(x_s) - f(x^*) \leq \frac{\mu_s}{2} L^2 + \left(\frac{1}{2\mu_s} - \frac{\alpha}{2}\right)\|x_s - x^*\|^2 - \frac{1}{2\mu_s}\|x_{s+1} - x^*\|^2.$$

Multiplicando la inecuación por s, obtenemos que

$$s(f(x_s) - f(x^*)) \leq \frac{L^2}{\alpha} + \frac{\alpha}{4}\left(s(s-1)\|x_s - x^*\|^2 - s(s+1)\|x_{s+1} - x^*\|^2\right),$$

Sumando ahora la inecuación resultante sobre $s = 1$ para $s = t$, y aplicando la inecuación de Jenssen 2.2, obtenemos el resultado final. □

Lo visto hasta ahora, y teniendo tanto la convexidad fuerte como la suavidad, nos permite una mejora drástica en el radio de convergencia. Sea $\kappa = \frac{\beta}{\alpha}$ para el número de condición de f.

Lema 4.17. *Sean $x, y \in X$, $x^+ = \Pi_X\Big(x - \frac{1}{\beta}\Delta f(x)\Big)$, y $g_X = \beta(x - x^+)$. Entonces tenemos la afirmación siguiente:*

$$f(x^+) - f(y) \leq g_X(x)^T(x - y) - \frac{1}{2\beta}\|g_X\|^2. \tag{4.26}$$

Este último lema, que no vamos a demostrar (ver [**4**]), nos da una observación clave, que puede mejorar el lema anterior para:

$$f(x^+) - f(y) \leq g_X(x)^T(x - y) - \frac{1}{2\beta}\|g_X\|^2 - \frac{\alpha}{2}\|x - y\|^2. \tag{4.27}$$

Teorema 4.18. *Sea f α-fuertemente convexa y β-suave en X. Entonces el método de descenso del gradiente proyectado con $\mu = \frac{1}{\beta}$ satisface para cualquier $k \geq 0$,*

$$\|x_{k+1} - x^*\|^2 \leq exp\Big(-\frac{k}{\kappa}\Big)\|x_0 - x^*\|^2.$$

Demostración. Usando la ecuación (4.27) con $y = x^*$, obtenemos que

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|x_k - \frac{1}{\beta}g_X(x_k) - x^*\|^2 \\
&= \|x_k - x^*\|^2 - \frac{2}{\beta}g_X(x_k)^T(x_k - x^*) + \frac{1}{\beta^2}\|g_X(x_k)\|^2 \\
&\leq \Big(1 - \frac{\alpha}{\beta}\Big)\|x_k - x^*\|^2 \\
&\leq \Big(1 - \frac{\alpha}{\beta}\Big)^t\|x_0 - x^*\|^2 \\
&\leq exp\Big(-\frac{k}{\kappa}\Big)\|x_0 - x^*\|^2,
\end{aligned}$$

lo que concluye nuestra prueba. □

A continuación, notemos que (4.23) y el lema que sigue a continuación son a veces referidos como la *coercitividad* del gradiente.

Lema 4.19. *Sea f β-suave y α-fuertemente convexa sobre $\mathbb{R}^n$. Entonces, para cualquier $x, y \in \mathbb{R}^n$, se tiene que*

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \frac{\alpha\beta}{\beta + \alpha}\|x - y\|^2 + \frac{1}{\beta + \alpha}\|\nabla f(x) - \nabla f(y)\|^2.$$

Demostración. Sea $\psi(x) = f(x) - \frac{\alpha}{2}\|x\|^2$. Por la definición de α-fuertemente convexa se tiene que ψ es convexa. Además, se puede ver que ψ es $(\beta - \alpha)$-suave por la prueba de (4.21) (y usando que ésto implica la suavidad de la función). Por tanto, usando (4.23) se obtiene que

$$(\nabla\psi(x) - \nabla\psi(y))^T(x - y) \geq \frac{1}{\beta - \alpha}\|\nabla\psi(x) - \nabla\psi(y)\|^2,$$

lo cual nos da el resultado que buscamos con sencillas operaciones. □

4.2. El método de Frank-Wolfe

Pasamos ahora a hablar sobre un algoritmo alternativo para minimizar una función convexa y β-suave sobre conjunto convexo y compacto $\mathcal{X}$. El gradiente condicional de descenso, introducido por Frank y Wolfe, realiza la siguiente actualización para $t \geq 1$, donde $(\eta_s)_{s\geq 1}$ es una secuencia fija,

$$y_t \in argmin_{y\in\mathcal{X}} \nabla f(x_t)^T y \tag{4.28}$$

$$x_{t+1} = (1-\eta_t)x_t + \eta_t y_t. \tag{4.29}$$

Es decir, el descenso del gradiente condicional va en la dirección óptima dado por el conjunto de restricciones X (para ello ver la siguiente figura). Desde una perspectiva

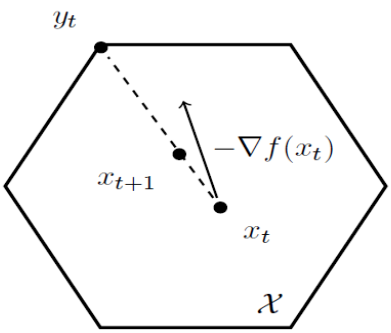

Figura 4.1: Ilustración del método de descenso del gradiente condicional

computacional, una propiedad clave de este método es que reemplaza el paso de proyección del método de descenso por gradiente proyectado por una optimización lineal sobre $\mathcal{X}$, lo cual, en algunos casos nos simplificará mucho el problema.

Pasemos entonces al análisis del método. Una de las mayores ventajas del método de descenso del gradiente condicional sobre el gradiente proyectado es que la primera puede adaptarse a la suavidad para cualquier norma arbitraria. Precisamente, sea f β-suave para alguna norma $\|.\|$, esto es que $\|\nabla f(x) - \nabla f(y)\|_* \leq \beta\|x-y\|$, donde la norma dual $\|.\|_*$ es definida como $\|g\|_* = sup_{x\in\mathbb{R}^n:\|x\|\leq 1} g^T$.

Teorema 4.20. *Sea f una función convexa y β-suave y sea alguna norma $\|.\|$, $R = sup_{x,y\in\mathcal{X}}\|x-y\|$, y $\eta_s = \frac{2}{s+1}$ para $s \geq 1$. Entonces, para cualquier $t \geq 2$,*

$$f(x_t) - f(x^*) \geq \frac{2\beta R^2}{t+1}.$$

Demostración. Las siguientes desigualdades son ciertas, usando respectivamente β-

suavidad, la definición de x_{s+1}, la definición de y_s y la convexidad de f:

$$\begin{aligned} f(x_{s+1}) - f(x_s) &\leq \nabla f(x_s)^T(x_{s+1} - x_s) + \frac{\beta}{2}\|x_{s+1} - x_s\|^2 \\ &\leq \eta_s \nabla f(x_s)^T(y_s - x_s) + \frac{\beta}{2}\eta_s^2 R^2 \\ &\leq \eta_s \nabla f(x_s)^T(x^* - x_s) + \frac{\beta}{2}\eta_s^2 R^2 \\ &\leq \eta_s(f(x^*) - f(x_s)) + \frac{\beta}{2}\eta_s^2 R^2. \end{aligned}$$

Reescribiendo la inecuación en término de $\delta_s = f(x_s) - f(x^*)$, obtenemos lo siguiente

$$\delta_{s+1} \leq (1 - \eta_s)\delta_s + \frac{\beta}{2}\eta_s^2 R^2.$$

Por inducción simple usando que $\eta_s = \frac{2}{s+1}$ concluye la prueba (notemos que la inicialización se realiza en el paso 2 con la desigualdad anterior con $\delta_2 \leq \frac{\beta}{2}R^2$). □

Además de ser "libre de proyecciones" y "libre de normas" (norma arbitraria), este método satisface quizás una propiedad aún más importante: *iteraciones escasas*. Más concretamente, consideremos la situación donde $\mathcal{X} \subset \mathbb{R}^n$ es un politopo, que es el casco convexo de un conjunto finito de puntos (estos puntos se denominan vértices de $\mathcal{X}$). Entonces, el teorema de Carathéodory afirma que cualquier punto $x \in \mathcal{X}$ puede ser escrito como una combinación convexa de al menos $n+1$ vértices de $\mathcal{X}$. Por otro lado, por la definición del descenso del gradiente condicional, se sabe que la t iteración de x_t puede ser escrita como una combinación lineal de t vértices (asumiendo que x_1 es un vértice). Gracias al radio de convergencia libre de la dimensión, es usual estar interesado en la manera en la que $t \ll n$, y por tanto, podemos ver que las iteraciones son muy escasas en la representación de sus vértices. Consecuentemente, obtenemos un interesante corolario sobre la propiedad de escasez junto con el radio de convergencia probado: funciones β-suaves en el simplex $\{x \in R_+^n \mid \sum_{i=1}^n x_i = 1\}$ siempre admite escasos minimizantes. Más concretamente aún, debe existir un punto x con solo t coordenadas distintas de cero y tal que $f(x) - f(x^*) = O(\frac{1}{t})$. Esto es lo mejor que uno puede esperar en general, que se puede ver con la función $f(x) = \|x\|_2^2$ ya que por Cauchy-Schwarz se tiene que $\|x\|_1 \leq \sqrt{\|x\|_0}\|x\|_2$.

4.3. Descenso geométrico

Hasta ahora, nuestros resultados dejan una brecha en el caso de la optimización de funciones β-suaves: el descenso del gradiente logra una complejidad de la predicción de $O(\frac{1}{\varepsilon})$ (respectivamente, en el caso de la convexidad fuerte un orden de $O(\kappa log(\frac{1}{\varepsilon}))$. En esta sección, cerraremos estas brechas con el ***método del descenso geométrico***. Históricamente, el primer método con una complejidad óptima de la predicción fue propuesto en el libro Nemirovski and Yudin [**11**]. Dicho método, inspirado en el gradiente conjugado, asume una predicción para calcular búsquedas de planos. En A.Nemirovski, (*O*rth-method for smooth convex optimization. Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika,2 ,1982) esta hipótesis fue suavizada por una predicción de la búsqueda de línea. Finalmente en Nesterov [**12**] se introdujo un

método óptimo que requiere una predicción de primer orden. El último algoritmo, llamado *descenso por gradiente acelerado de Nesterov*, ha sido el método más importante y óptimo para funciones β-suaves hasta el día de hoy (el cual, describiremos y analizaremos en la siguiente sección). Como veremos, la intuición detrás del descenso acelerado de Nesterov no es del todo transparente, lo cual motiva la presente sección, ya que el descenso geométrico tiene una interpretación geométrica inspirada el método del elipsoide.

Nos centraremos en la optimización de funciones convexas β-suaves y α-fuertemente convexas con restricciones, y probaremos la complejidad de la predicción de un orden de $O(\sqrt{\kappa}log(\frac{1}{\varepsilon}))$, reduciendo así la complejidad del descenso del gradiente por un factor $\sqrt{\kappa}$. Esta mejora es muy relevante para aplicaciones de *machine learning*. Consideremos entonces un ejemplo de un problema de regresión logística: esto es un problema de una función β-suave y α-fuertemente convexa, cuyo párametro α es del orden de una constante numérica, pero con párametro β cuyo inverso puede ser del orden del tamaño de la muestra. En este caso, κ puede ser del orden del tamaño de la muestra, y con un radio mucho más rápido por un factor $\sqrt{\kappa}$.

También veremos que esta mejora del radio para funciones objetivos β-suaves y α-fuertemente convexas implica un radio casi ótpimo del orden de $O(log(1/\varepsilon)/\sqrt{\varepsilon})$ para el caso de funciones β-suaves, ya que uno puede evaluar un descenso geométrico en la función $x \mapsto f(x) + \varepsilon\|x\|^2$.

4.3.1. Preliminares

Comencemos con algunas notaciones. Sea $B(x, r^2) = \{y \in R^n \,|\, \|y - x\|^2 < R^2\}$ y

$$x^+ = x - \frac{1}{\beta}\nabla f(x), \quad x^{++} = x - \tfrac{1}{\alpha}\nabla f(x).$$

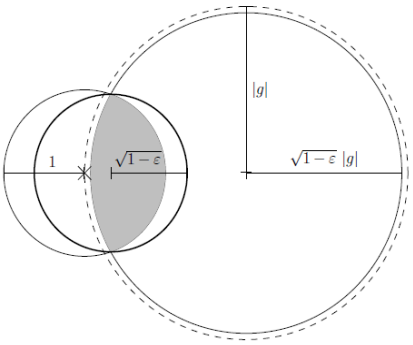

Figura 4.2: Bola encerrada

Reescribiendo la definición de convexidad fuerte, obtenemos lo siguiente

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\alpha}{2}\|y - x\|^2$$

que es equivalente a

$$\frac{\alpha}{2}\|y - x + \frac{1}{\alpha}\nabla f(x)\|^2 \leq \frac{\|\nabla f(x)\|^2}{2\alpha} - (f(x) - f(y)),$$

y uno obtiene una bola encerrada para el minimizante de f con el 0 y 1 orden de información de x:

$$x^* \in B\left(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2} - \frac{2}{\alpha}(f(x) - f(x^*))\right).$$

Además, recordemos que por la suavidad se tiene que $f(x^+) \leq f(x) - \frac{1}{2\beta}\|\nabla f(x)\|^2$ lo que permite encoger la bola anterior por un factor de $1 - \frac{1}{\kappa}$ y obtener

$$x^* \in B\left(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2}(1 - \frac{1}{\kappa}) - \frac{2}{\alpha}(f(x^+) - f(x^*))\right). \tag{4.30}$$

Eso nos sugiere lo siguiente: asumimos que se tiene que la bola $A := B(x, R^2)$ y la bola $B\left(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right)\right)$. Siempre y cuando el radio de B sea una fracción del radio de A, luego se puede iterar el procedimiento reemplazando A por B, lo que lleva a una convergencia lineal más rápida.

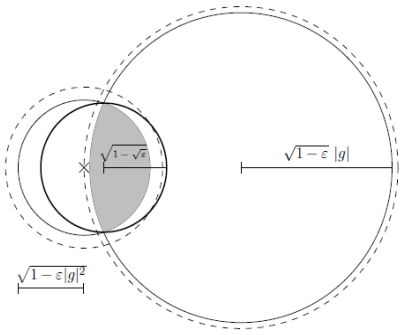

Figura 4.3: Dos bolas encerradas

Evaluar la velocidad a la que se contrae el radio es un cálculo sencillo: para cualquier $g \in \mathbb{R}^n, \varepsilon \in (0, , 1)$, existe $x \in \mathbb{R}^n$ tal que

$$B(0,1) \cap B(g, \|g\|^2(1-\varepsilon)) \subset B(x, 1-\varepsilon).$$

(Ver Figura 4.3.1.)

Por tanto, vemos que en la estrategia de la bola anterior, el radio al cuadrado de la bola circundante para x se reduce a un factor de $\frac{1}{\kappa}$ en cada iteración, igualándose así al radio de convergencia del método de descenso del gradiente.

4.3.2. El método

Sea $x_0 \in \mathbb{R}^n$, $c_0 = x_0^{++}$ y $R_0^2 = \left(1 - \frac{1}{\kappa}\right)\frac{\|\nabla f(x_0)\|^2}{\alpha^2}$. Para cualquier $t \geq 0$ sea

$$x_{t+1} = argmin_{x \in \{(1-\lambda)c_t + \lambda x_t^+, \lambda \in \mathbb{R}\}} f(x),$$

y c_{t+1} (respectivamente R^2_{t+1}) el centro (respectivamente el radio al cuadrado) de la bola dada por la prueba dada en el Lema 4.22 (que veremos en el siguiente capítulo) que contiene

$$B\left(c_t, R_t^2 - \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2\kappa}\right) \cap B\left(x_{t+1}^{++}, \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2}\Big(1 - \frac{1}{\kappa}\Big)\right).$$

donde uno puede usar las siguientes fórmulas para c_{t+1} y R^2_{t+1}. Si $\frac{|\nabla f(x_{t+1})|^2}{\alpha^2} < \frac{R_t^2}{2}$ entonces se puede tomar $c_{t+1} = x_{t+1}^{++}$ y $R^2_{t+1} = \frac{\nabla f(x_{t+1})|^2}{\alpha^2}\Big(1 - \frac{1}{\kappa}\Big)$. Por otro lado, si $\frac{|\nabla f(x_{t+1})|^2}{\alpha^2} \geq \frac{R_t^2}{2}$ entonces se toma

$$c_{t+1} = c_t + \frac{R_t^2 + |x_{t+1} - c_t|^2}{2|x_{t+1}^{++} - c_t|^2}(x_{t+1}^{++} - c_t,$$

$$R^2_{t+1} = R_t^2 - \frac{|\nabla f(x_{t+1})|^2}{\alpha^2\kappa} - \left(\frac{R_t^2 + \|x_{t+1} - c_t\|^2}{2\|x_{t+1}^{++} - c_t\|}\right)^2.$$

Teorema 4.21. *Para cualquier $t \geq 0$, se tiene que*

$$x^* \in B(c_t,\, R_t^2),\; R^2_{t+1} \leq \Big(1 - \frac{1}{\sqrt{\kappa}}\Big)R_t^2$$

, y por tanto

$$\|x_t - c_t\|^2 \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)^t R_0^2.$$

Demostración. Probaremos algo más fuerte por inducción, de manera que, para cada $t \geq 0$, tenemos

$$x^* \in \left(c_t, R_t^2 - \frac{2}{\alpha}\Big(f(x_t^+) - f(x^*)\Big)\right).$$

El caso de $t = 0$ se tiene inmediatamente por (4.30). Por tanto, asumamos que es cierto para algún $t \geq 0$. Entonces usando que $f(x_{t+1}^+) \leq f(x_{t+1}) - \frac{1}{2\beta}\|\nabla f(x_{t+1})\|^2 \leq f(x_t^+ - \frac{1}{2\beta}\|\nabla f(x_{t+1})\|^2$, uno tiene que

$$x^* \in B\left(c_t, R_t^2 - \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2\kappa} - \frac{2}{\alpha}\Big(f(x_{t+1}^+) - f(x^*)\Big)\right).$$

Además, por (4.30) también se tiene que

$$B\left(x_{t+1}^{++}, \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2}\Big(1 - \frac{1}{\kappa}\Big) - \frac{2}{\alpha}\Big(f(x_{t+1}^+) - f(x^*)\Big)\right).$$

Por tanto, sólo queda observar que el radio al cuadrado de la bola dado por el Lema 4.22 que encierra la intersección de las dos bolas anteriores es más pequeño que $\Big(1 - \frac{1}{\sqrt{\kappa}}\Big)R_t^2 - \frac{2}{\alpha}f(x_{t+1}^+) - f(x^*))$. Aplicando el lema 4.21 después de llevar c_t al origen y adaptando las distancias por R_t. Tomando $\varepsilon = \frac{1}{\kappa}$, $g = \frac{\|\nabla f(x_{t+1})\|}{\alpha}$,$\gamma = \frac{2}{\alpha}\Big(f(x_{t+1}^+ - f(x^*)\Big)$ y $a = x_{t+1}^{++} - c_t$. La línea de paso de búsqueda del algoritmo implica que $\nabla f(x_{t+1})^T(x_{t+1} - c_t) = 0$ y por lo tanto, $\|a\| = \|x_{t+1}^{++} - c_t\| \geq \|\frac{\nabla f(x_{t+1})}{\alpha}\| = g$ y aplicando el Lema 4.21 obtenemos el resultado deseado. □

4.3.3. Aceleración

En la sección de Preliminares obviamos la ocasión de ver la bola $A = B(x, R^2)$ como intersección de las bolas dadas en la ecuación 4.30 y, por tanto, el nuevo valor $f(x)$ también podría usarse para reducir el radio de esas bolas anteriores (una cosa a tener en cuenta es que el valor $f(x)$ debe ser más pequeño que los valores usados para construir dichas bolas. Ésto podría mostrar que el óptimo está de hecho contenido en la bola $B(x, R^2 - \frac{1}{\kappa}\|\nabla f(x)\|^2)$. Tomando intersección con la bola $B\Big(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2}\Big(1 - \frac{1}{\kappa}\Big)\Big)$ permitiría obtener una nueva bola con un radio reducido por el factor $1 - \frac{1}{\sqrt{\kappa}}$ (en vez de $1 - \frac{1}{\kappa}$): de hecho, para cualquier $g \in \mathbb{R}^n$, $\varepsilon \in (0, 1)$, $\exists x \in \mathbb{R}^n$ tal que

$$B(0, 1 - \varepsilon\|g\|^2) \cap B(g, \|g\|^2(1 - \varepsilon)) \subset B(x, 1 - \sqrt{\varepsilon}).$$

(Ver Figura 4.3.1.) Por lo tanto, sólo queda tratar con la advertencia mencionada anteriormente, que hacemos a través de una línea de búsqueda. A su vez, dicha línea de búsqueda puede cambiar la nueva bola (4.30), y para tratar con esto necesitaremos el siguiente lema de la inclusión anterior.

Lema 4.22. *Sea $a \in \mathbb{R}^n$, $\varepsilon \in (0, 1)$ y $g \in \mathbb{R}_+$. Supongamos que $\|a\| \geq g$. Entonces existe $c \in \mathbb{R}^n$ tal que, para cualquier $\gamma \geq 0$,*

$$B(0, 1 - \varepsilon g^2 - \gamma) \cap B(a, g^2(1 - \varepsilon) - \gamma) \subset B(c, 1 - \sqrt{\varepsilon} - \gamma).$$

La demostración la podemos encontrar en [**4**].

4.4. Descenso por gradiente acelerado de Nesterov

Pasamos ahora a hablar del *método original de Nesterov* que alcanza la complejidad de la predicción óptima para una optimización convexa suave. Daremos los detalles del método tanto para el caso de optimización fuertemente convexa como para el caso contrario. Para ello, nos referiremos a *Su et al. [2014]* [**13**] para una interpretación más reciente del método en términos de ecuaciones diferenciales, y a *Allen-Zhou y Orechia [2014]* [**14**] para su relación con *'mirror descent'*.

El método del gradiente de descenso acelerado de Nesterov (que se puede ver en la Figura (4.4), puede ser descrito de la siguiente manera: Empezamos con un punto arbitrario incial $x_1 = y_1$ para luego iterar de la siguiente manera:

$$\begin{aligned} y_{t+1} &= x_t \frac{1}{\beta}\nabla f(x_t), \\ x_{t+1} &= \left(1 + \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right) y_{t+1} - \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} y_t. \end{aligned}$$

Teorema 4.23. *Sea f α-fuertemente convexa y β-suave, entonces el método del gradiente de descenso acelerado de Nesterov satisface la siguiente inecuación:*

$$f(y_t) - f(x^*) \leq \frac{\alpha + \beta}{2}\|x_1 - x^*\|^2 exp\Big(-\frac{t - 1}{\sqrt{\kappa}}\Big).$$

Demostración. Definimos las funciones cuadráticas α-fuertemente convexas como ϕ_s, $s \geq 1$ por inducción como:

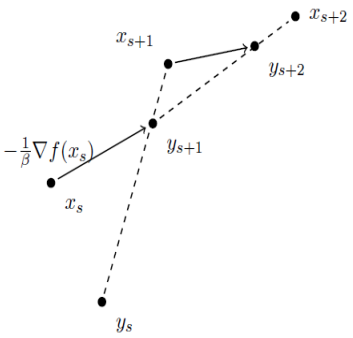

Figura 4.4: Gradiente de descenso acelerado de Nesterov

$$\phi_1(x) = f(x_1) + \frac{\alpha}{2}\|x - x_1\|^2,$$

$$\phi_{s+1}(x) = \left(1 - \frac{1}{\sqrt{\kappa}}\right)\phi_s(x) + \frac{1}{\sqrt{\kappa}}\left(f(x_s) + \nabla f(x_s)^T(x - x_s) + \frac{\alpha}{2}\|x - x_s\|^2\right) \quad (4.31)$$

. Intuitivamente ϕ_s se convierte en una aproximación más fina de f en el siguiente sentido:

$$\phi_{s+1}(x) \leq f(x) + \left(1 - \frac{1}{\sqrt{\kappa}}\right)^s (\phi_1(x) - f(x)). \quad (4.32)$$

Esta última inecuación puede ser probada por inducción, usando el hecho de que por la α-fuerte convexidad se tiene que

$$f(x_s) + \nabla f(x_s)^T(x - x_s) + \frac{\alpha}{2}\|x - x_s) + \frac{\alpha}{2}\|x - x_s\|^2 \leq f(x).$$

La ecuación (4.32) por sí misma no nos dice mucho, pero para que sea de utilidad, qué tan por debajo de f está ϕ_s. La siguiente inecuación responde a tal:

$$f(y_s) \leq \min_{x \in \mathbb{R}^n} \phi_s(x). \quad (4.33)$$

El resto de la prueba se ciñe a mostrar que la inecuación anterior es cierta, pero primero, veamos cómo combinar las dos inecuaciones anteriores vistas para obtener la tasa dada por el teorema (usaremos que por la β-suavidad tenemos que $f(x) - f(x^*) \leq \frac{\beta}{2}\|x - x^*\|^2$):

$$\begin{aligned} f(y_t) - f(x^*) &\leq \phi_t(x^*) - f(x^*) \\ &\leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)^{t-1} (\phi_1(x^*) - f(x^*)) \\ &\leq \frac{\alpha + \beta}{2}\|x_1 - x^*\|^2 \left(1 - \frac{1}{\sqrt{\kappa}}\right)^{t-1}. \end{aligned}$$

Ahora probaremos la inecuación (4.33) por inducción (tengamos en cuenta que es cierto para $s = 1$ por lo que $x_1 = y_1$). Sea $\phi_s^* = \text{mín}_{x\in\mathbb{R}^n}\, \phi_s(x)$. Usando la definición de y_{s+1}, la β-suavidad, convexidad y la hipótesis inductiva, uno tiene lo siguiente:

$$\begin{aligned} f(y_{s+1}) &\leq f(x_s) - \frac{1}{2\beta}\|\nabla f(x_s)\|^2 \\ &= \left(1 - \frac{1}{\sqrt{\kappa}}\right) f(y_s) + \left(1 - \frac{1}{\sqrt{\kappa}}\right)(f(x_s) - f(y_s)) + \frac{1}{\sqrt{\kappa}} f(x_s) - \frac{1}{2\beta}\|\nabla f(x_s)\|^2 \\ &\leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)\phi_s^* + \left(1 - \frac{1}{\sqrt{\kappa}}\right)\nabla f(x_s)^T(x_s - y_s) + \frac{1}{\sqrt{\kappa}} f(x_s) - \frac{1}{2\beta}\|\nabla f(x_s)\|^2 \end{aligned}$$

Por lo que ahora tenemos que mostrar que

$$\phi_{s+1}^* \geq \left(1 - \frac{1}{\sqrt{\kappa}}\right)\phi_s^* + \left(1 - \frac{1}{\sqrt{\kappa}}\right)\nabla f(x_s)^T(x_s - y_s) + \frac{1}{\sqrt{\kappa}} f(x_s) - \frac{1}{2\beta}\|\nabla f(x_s)\|^2. \tag{4.34}$$

Para probar esta inecuación, tenemos que entender mejor las funciones ϕ_s. Hemos de notar que $\nabla^2\phi_s(x) = \alpha I_n$ y por lo tanto ϕ_s tiene que ser de la siguiente forma:

$$\phi_s(x) = \phi_s^* + \frac{\alpha}{2}\|x - v_s\|^2,$$

para algún $v_s \in \mathbb{R}^n$. Observemos ahora que derivando (4.21) y usando la forma de ϕ_s anterior, uno obtiene que

$$\nabla\phi_{s+1}(x) = \alpha\left(1 - \frac{1}{\sqrt{\kappa}}\right)(x - v_s) + \frac{1}{\sqrt{\kappa}}\nabla f(x_s) + \frac{\alpha}{\sqrt{\kappa}}(x - x_s).$$

En particular, ϕ_{s+1} se minimiza por definición como v_{s+1} que ahora se puede definir por inducción usando la identidad de arriba:

$$v_{s+1} = \left(1 - \frac{1}{\sqrt{\kappa}}\right)v_s + \frac{1}{\sqrt{\kappa}}x_s - \frac{1}{\alpha\sqrt{\kappa}}\nabla f(x_s). \tag{4.35}$$

Teniendo en cuenta la forma de ϕ_s y ϕ_{s+1}, y además de la definición de 4.20, uno consigue la siguiente identidad evaluando x_s en ϕ_{s+1}:

$$\phi_{s+1}^* + \frac{\alpha}{2}\|x_s - v_{s+1}\|^2 = \left(1 - \frac{1}{\sqrt{\kappa}}\right)\phi_s^* + \frac{\alpha}{2}\left(1 - \frac{1}{\sqrt{\kappa}}\right)\|x_s - v_s\|^2 + \frac{1}{\sqrt{\kappa}} f(x_s). \tag{4.36}$$

Gracias a (4.35), uno tiene que

$$\begin{aligned} \|x_s - v_{s+1}\|^2 &= \left(1 - \frac{1}{\sqrt{\kappa}}\right)^2\|x_s - v_s\|^2 + \frac{1}{\alpha^2\kappa}\|\nabla f(x_s)\|^2 \\ &- \frac{2}{\alpha\sqrt{\kappa}}\left(1 - \frac{1}{\sqrt{\kappa}}\right)\nabla f(x_s)^\top(v_s - x_s), \end{aligned}$$

que junto a (4.36)

$$\begin{aligned}\phi^*_{s+1} &= \left(1-\frac{1}{\sqrt{\kappa}}\right)\phi^*_s + \frac{1}{\sqrt{\kappa}}f(x_s) + \frac{\alpha}{2\sqrt{\kappa}}\left(1-\frac{1}{\sqrt{\kappa}}\right)\|x_s - v_s\|^2 \\ &- \frac{1}{2\beta}\|\nabla f(x_s)\|^2 + \frac{1}{\sqrt{\kappa}}\left(1-\frac{1}{\sqrt{\kappa}}\right)\nabla f(x_s)^\top (v_s - x_s).\end{aligned}$$

Por último, mostramos por inducción que $v_s - x_s = \sqrt{\kappa}(x_s - y_s)$, que concluye la prueba de (4.34) y , por tanto, también concluye la prueba del teorema:

$$\begin{aligned}v_{s+1} - x_{s+1} &= \left(1-\frac{1}{\sqrt{\kappa}}\right)v_s + \frac{1}{\sqrt{\kappa}}x_s - \frac{1}{\alpha\sqrt{\kappa}}\nabla f(x_s) - x_{s+1} \\ &= \sqrt{\kappa}x_s - (\sqrt{\kappa}-1)y_s - \frac{\sqrt{\kappa}}{\beta}\nabla f(x_s) - x_{s+1} \\ &= \sqrt{\kappa}y_{s+1} - (\sqrt{\kappa}-1)y_s - x_{s+1} \\ &= \sqrt{\kappa}(x_{s+1} - y_{s+1}),\end{aligned}$$

donde la primera igualdad la obtenemos de (4.35), la segunda de la hipótesis inductiva, la tercera de la definición de y_{s+1} y la última de la definición de x_{s+1}. □

Pasamos entonces a tratar el caso en el que la optimización sea para funciones f que son α-suaves. En primer lugar, mostremos cómo adaptar el método de descenso del gradiente acelerado de Nesterov para el caso $\alpha = 0$, usando para ello una combinación variable en el tiempo de los elementos de la sucesión primaria (y_t). Definamos primero las siguientes sucesiones:

$$\lambda_0 = 0, \quad \lambda_t = \tfrac{1+\sqrt{1+4\lambda^2_{t-1}}}{2}, \quad \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}$$

Pues bien, ahora el algoritmo es simplemente definido por las siguientes ecuaciones, con $x_1 = y_1$ un punto inicial arbitrario:

$$\begin{aligned}y_{t+1} &= x_t - \frac{1}{\beta}\nabla f(x_t), \\ x_{t+1} &= (1-\gamma_s)y_{t+1} + \gamma_t y_t.\end{aligned}$$

Lema 4.24. *Sea el problema con restricciones:*

$$\begin{aligned}&min. f(x) \\ &s.t.\ x \in \mathcal{X}\end{aligned}$$

Sean $x, y \in \mathcal{X}$,$x^+ = \Pi_{\mathcal{X}}\left(x - \frac{1}{\beta}\nabla f(x)\right)$, *y* $g_{\mathcal{X}}(x) = \beta(x - x^+)$. *Entonces lo siguiente es cierto:*

$$f(x^+) - f(y) \leq g_{\mathcal{X}}(x)^\top (x-y) - \frac{1}{2\beta}\|g_{\mathcal{X}}(x)\|^2.$$

Para la prueba ver en Bubeck (2015) [**4**].

Teorema 4.25. *Sea f una función β-suave y convexa, entonces el método del gradiente de descenso acelerado de Nesterov satisface*

$$f(y_t) - f(x^*) \leq \frac{2\beta\|x_1 - x^*\|^2}{t^2}.$$

*Nos guiaremos para ello de la prueba de Beck y Teobulle [**3**]. También nos referiremos a Tseng (On accelerated proximal gradient methods for convex-convave optimization, 2008.) para una prueba con tamaños de paso más simples.*

Demostración. Usando la versión sin restricciones del Lema 4.24 para el caso con restricciones, obtenemos que:

$$\begin{aligned} f(y_{s+1}) - f(y_s) &\leq \nabla f(x_s)^\top (x_s - y_s) - \frac{1}{2\beta}\|\nabla f(x_s)\|^2 \\ &= \beta(x_s - y_{s+1})^\top (x_s - y_s) \frac{\beta}{2}\|x_s - y_{s+1}\|^2. \end{aligned} \tag{4.37}$$

De manera análoga tenemos que

$$f(y_{s+1}) - f(x^*) \leq \beta(x_s - y_{s+1})^\top (x_s - x^*) - \frac{\beta}{2}\|x_s - y_{s+1}\|^2. \tag{4.38}$$

Pues bien, multiplicando (4.37) por $(\lambda_s - 1)$ y añadiendo el resultado a (4.38), uno consigue que $\delta_s = f(y_s) - f(x^*)$,

$$\lambda_s\delta_{s+1} - (\lambda_s - 1)\delta_s \leq \beta(x_s - y_{s+1})^\top (\lambda_s x_s - (\lambda_s - 1)y_s - x^*) - \frac{\beta}{2}\|x_s - y_{s+1}\|^2.$$

Multiplicando esta última inecuación por λ_s y usando esto por la definición de $\lambda_{s-1}^2 = \lambda_s^2 - \lambda_s$, así como la identidad elemental $2a^T b - \|a\|^2 = \|b\|^2 - \|b - a\|^2$, obtenemos lo siguiente:

$$\begin{aligned} &\lambda_s^2\delta_{s+1} - \lambda_{s-1}^2\delta_s \leq \\ &\leq \frac{\beta}{2}\left(2\lambda_s(x_s - y_{s+1})^\top (\lambda_s x_s - (\lambda_s - 1)y_s - x^*) - \|\lambda_s(y_{s+1} - x_s)\|^2\right) \\ &= \frac{\beta}{2}\left(\|\lambda_s x_s - (\lambda_s - 1)y_s - x^*\|^2 - \|\lambda_s y_{s+1} - (\lambda_s - 1)y_s - x^*\|^2\right). \end{aligned} \tag{4.39}$$

Se puede observar que, por definición, tenemos

$$\begin{aligned} x_{s+1} &= y_{s+1} + \gamma_s(y_s - y_{s+1}) \\ \Leftrightarrow \lambda_s y_{s+1} &= \lambda_s y_{s+1} + (1 - \lambda_s)(y_s - y_{s+1}) \\ \Leftrightarrow \lambda_s y_{s+1} - (\lambda_{s+1} - 1)y_{s+1} &= \lambda_s y_{s+1} - (\lambda_s - 1)y_s. \end{aligned} \tag{4.40}$$

Juntando (4.39) con (4.40) uno obtiene con $u_s = \lambda_s x_s - (\lambda_s - 1)y_s - x^*$,

$$\lambda_s^2\delta_{s+1} - \lambda_{s-1}^2\delta_s^2 \leq \frac{\beta}{2}\left(\|u_s\|^2 - \|u_{s+1}\|^2\right).$$

Sumando estas inecuaciones desde $s = 1$ hasta $s = t - 1$ conseguimos:

$$\delta_t \leq \frac{\beta}{2}\lambda_{t-1}^2\|u_1\|^2.$$

Por inducción, es fácil ver que $\lambda_{t-1} \geq \frac{t}{2}$ lo que concluye nuestra prueba. □

Capítulo 5

Métodos alternativos

Consideraremos un nuevo método algorítmico denominado *ISTA* para resolver problemas inversos lineales que surgen en el procesamiento de señales o imágenes. Este tipo de métodos, los cuales pueden ser visto como una extensión del método clásico del gradiente, resultan atractivos debido a su simplicidad y por tanto, son adecuados para la resolución de problemas de gran magnitud incluso con datos de matriz densa. Sin embargo, tales métodos son también conocidos por su lenta convergencia. En esta tesitura, hablaremos entonces de un nuevo método más rápido (en lo que a convergencia se refiere) denominado *FISTA*, el cual, preserva la simplicidad computacional del método ISTA pero con un radio global de convergencia el cual ha demostrado ser significativamente mejor, tanto prácticamente como teóricamente. Los resultados numéricos iniciales (prometedores) para desvanecer las imágenes borrosas basadas en ondículas demuestran la capacidad de FISTA, que se presenta más rápida que la de ISTA en varios órdenes de grande magnitud.

5.1. ISTA (Iterative Shrinkage-Thresholding Algorithm)

5.1.1. Introducción

Los problemas lineales inversos aumentan en un gran rango de aplicaciones tales como astrofísica, procesamiento de imágenes y señales, inferencia estadística, y óptica, entre otros. La naturaleza interdisciplinaria de los problemas inversos es evidente a través de una vasta literatura que incluye a un gran número de desarrollos matemáticos y algorítmicos. Un problema básico lineal e inverso lleva a estudiar un sistema lineal de la siguiente forma

$$Ax = b + w \tag{5.1}$$

donde $A \in \mathbb{R}^{m \times n}$ con $b \in \mathbb{R}^m$ conocido, w es una perturbación y x es la imagen/señal desconocida que queremos estimar. En los problemas de imagen borrosa, cuyo tamaño se asume igual que el de b ($m = n$); tanto b como x se forman apilando las columnas de sus correspondientes imágenes bidimensionales. En este tipo de aplicaciones, la matriz A describe el *operador de desenfoque*, que en el caso de desenfoques espalciamentes invariantes, representa un operador de convolución de dos dimensiones. El problema de estimar x a partir de la borrosidad y la imagen perturbada b, se conoce como *problema de borrado de imagen*. Un enfoque clásico del problema

(5.1) es el de mínimos cuadrados *(least squares)*, en el que se elige un estimador para minimizar el error de datos:

$$(LS) : \hat{x}_{LS} = \text{argmin}_{\,x} \|Ax - b\|^2.$$

Cuando $m = n$ y A no es singular, el estimador LS es la solución $A^{-1}B$. En muchas aplicaciones, es frecuente el caso en el que A está mal condicionada, por lo que, en dichos casos, la solución LS tiene (normalmente) una norma muy grande, y por lo tanto, carece de sentido. Para solver esto, aparece la idea básica de regularización que es reemplazar el problema mal condicionado por un problema similar pero bien condicionado cuya solución se aproxima a la solución requerida. Una de las técnicas más populares es la *regularización de Tikhonov* en la que añadimos una penalización cuadrática:

$$(T) : \hat{x}_{TIK} = \text{argmin}_{\,x} \{\|Ax - b\|^2 + \lambda \|Lx\|^2\} \tag{5.2}$$

El segundo término del problema de minimización de arriba es un término que controla la norma (o seminorma) de la solución. El parámetro de regularización $\lambda > 0$ proporciona una compensación entre la adhesión a las mediciones y la perturbación. Las elecciones para L suelen ser la identidad o una matriz que se aproxima al operador derivado de primer o segundo orden.

Otro método de regularización, que ha atraído un gran interés en el procesamiento de señales es el de ℓ_1-regularición en el que se busca encontrar la solución de

$$\min_x \{F(x) \equiv \|Ax - b\|^2 + \lambda \|x\|_1\}. \tag{5.3}$$

En las aplicaciones de eliminación de imágenes borrosas, y en particular en los métodos de restauración basados en *wavelet*. A suele ser elegida como $A = RW$, donde R es la *matriz borrosa* y W contiene una base de wavelet. El vector x contiene los coeficientes de la imagen desconocida.

Nota 5.1. *Las **wavelets** son funciones que satisfacen ciertos requerimientos matemáticos y son utilizados para la representación de datos o de otras funciones. Son muy adecuadas para aproximación de datos con variaciones o con discontinuidades abruptas.*

La filosofía subyacente en tratar con el criterio de regularización de la norma ℓ_1 es que la mayoría de las imágenes tienen una representación escasa en el dominio de las wavelets. El problema de optimización (5.4) se puede convertir en un problema de cono de segundo orden (ver [**5**]). Sin embargo, en la mayoría de las aplicaciones, el problema no es solo a gran escala (puede alcanzar millones de variables determinantes) pero también involucra datos de la matriz muy densos, que suelen imposibilitar el uso y la enorme ventaja de métodos sofisticados de puntos interiores. Esto motivó la búsqueda de simplificar los algoritmos basados en el gradiente, para resolver el problema (5.3), donde el dominante esfuerzo computacional es una multiplicación de matriz-vector relativamente sencilla involucrando A y A^t.

Uno de los métodos más conocidos para resolver dicho problema es **ISTA** (**iterative shrinkage-thresholding algorithms**), donde cada iteración involucra una multiplicación de matriz-vector (que involucra a A y A^t) seguido de un paso de contracción. Dicho método, y paso principal, consiste en el siguiente:

$$x_{k+1} = \mathcal{T}_{\lambda t}\big(x_k - 2tA^t(Ax_k - b)\big) \tag{5.4}$$

donde t es un tamaño de paso adecuado y $\mathcal{T}_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ es el operador de contracción definido como

$$\mathcal{T}_\alpha(x)_i = (|x_i| - \alpha)_+ sg(x_i). \tag{5.5}$$

Este algoritmo puede ser remontado hasta el esquema iterativo de hacia delante y hacia detrás, introducido en [**8**] y [**9**] dentro del marco general de los métodos de división. Otra interesante contribución reciente incluye muchos resultados generales de convergencia para la sucesión (x_k) producida por algoritmos proximales hacia atrás y hacia delante bajo varias condiciones y bajo importantes ajustes de problemas inversos lineales.

Recientemente, otros investigadores han estado trabajando en algoritmos alternativos que podrían acelerar el rendimiento de ISTA, como FISTA, de manera que, confían en que estos métodos podrían calcular la siguiente iteración basándose no sólo en la anterior, sino en dos o más iteraciones calculadas previamente. Otros de los métodos que se proponen es el **TWIST**, que no es más que un interesante ISTA de dos pasos que, bajo algunas hipótesis sobre los datos del problema y los parámetros elegidos apropiadamente que definen el algoritmo, ha demostrado que converge en un minimizante de la función objetivo de la forma:

$$\|Ax - b\|^2 + \phi(x) \tag{5.6}$$

donde ϕ es una función convexa β-suave. La efectividad de TWIST como un método más rápido que ISTA fue demostrado experimentalmente en varios problemas inversos lineales. Otra línea de análisis de la aceleración de ISTA para la misma clase de problemas que el anterior, fue considerado en [**10**] utilizando técnicas de optimización del subespacio secuencial y confiando en generar la siguiente iteración minimizando una función sobre un supespacio afín abarcado por dos o más iteraciones previas y el gradiente corriente. La aceleración obtenida con este enfoque se ha demostrado a través de experimentos numéricos para eliminar los problemas de aplicación. Para ambos casos, no se ha establecido la tasa global de convergencia no asintótica.

5.1.2. Descripción del método

La idea básica de ISTA no es más que construir en cada iteración una regularización de la parte de función diferenciable linealizada en la función objetivo. Para nuestro análisis, consideraremos la siguiente formulación general, la cual extiende el problema de formulación (5.3) :

$$\text{mín}\{F(x) \equiv f(x) + g(x) : x \in \mathbb{R}^n\}, \tag{5.7}$$

tomando las siguientes hipótesis:

- $g : \mathbb{R}^n \to \mathbb{R}$ es una función continua y convexa, que posiblemente no sea β-suave.
- $f : \mathbb{R}^n \to \mathbb{R}$ es una función convexa, β-suave del tipo $C^{1,1}$, es decir, continua y diferenciable con gradiente Lipschitziano $\nabla L(f)$, es decir, dados $x, y \in \mathbb{R}^n$:

  $$\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|$$

  donde $\|\cdot\|$ denota la norma euclídea y $L(f) > 0$ es la constante lipschitziana de ∇f.

- El problema anterior (5.7) es resoluble, es decir, $X_* := \operatorname{argmin} F \neq \emptyset$, y para $x^* \in X_*$ definimos $F_* := F(x^*)$.

Ejemplo 5.2. *Cuando $g(x) \equiv 0$, el problema (5.7) es el problema general de minimización convexa sin restricciones.*

Ejemplo 5.3. *El problema de regularización ℓ_1 (5.3) es un caso especial de (5.7) sustituyendo $f(x) = \|Ax - b\|^2$,$g(x) = \|x\|_1$. La constante lipschitziana del gradiente ∇f es $L(f) = 2\lambda_{\text{máx}}(A^t A)$.*

Consideramos el siguiente modelo aproximado: para cualquier $L > 0$, sea $F(x) := f(x) + g(x)$ la aproximación cuadrática, y dado un punto y:

$$Q_L(x, y) := f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x), \tag{5.8}$$

que admite un minimizante único

$$p_L(y) := \operatorname{argmin}\{Q_L(x, y) : x \in \mathbb{R}^n\}. \tag{5.9}$$

El álgebra por tanto nos muestra que (ignorando constantes en términos de y):

$$p_L(y) = \operatorname{argmin}_x \left\{ g(x) + \frac{L}{2}\|x - \Big(y - \frac{1}{L}\nabla f(y)\Big)\|^2 \right\}.$$

De manera que, es fácil ver que el paso básico de ISTA para el problema (5.7) se reduce a

$$x_k = p_L(x_{k-1}).$$

Previo al análisis de ISTA, vamos a dar una serie de resultados claves que serán crucial para el análisis de éste. Para ello, primero necesitamos recordar el primer lema, que es la siguiente y bien conocida propiedad para la funciones β-suaves de clase $C^{1,1}$.

Lema 5.4. *Para cualquier $y \in \mathbb{R}^n$, se tiene que $z = p_L(y)$ si y sólo si existe $\gamma(y) \in \partial g(z)$, el subdiferencial de g, tal que*

$$\nabla f(y) + L(z - y) + \gamma(y) = 0. \tag{5.10}$$

Demostración. El resultado es inmediato gracias a las condiciones de optimalidad para el problema convexo fuerte (5.9). □

Lema 5.5. *Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función continua y diferenciable con gradiente continuo y constante de Lipschitz $L(f)$. Entonces, para cualquier $L \geq L(f)$,*

$$f(x) \leq f(y)\langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 \tag{5.11}$$

para cualquier $x, y \in \mathbb{R}^n$.

Demostración. Es una consecuencia inmediata del lema anterior. □

Lema 5.6. *Sea $y \in \mathbb{R}^n$ y $L > 0$ tales que*

$$F(p_L(y)) \leq Q(p_L(y), y). \tag{5.12}$$

Entonces, para cualquier $x \in \mathbb{R}^n$, se tiene que:

$$F(x) - F(p_L(y)) \geq \frac{L}{2}\|p_L(y) - y\|^2 + L\langle y - x, p_L(y) - y\rangle. \tag{5.13}$$

Demostración. De (5.12), tenemos que

$$F(x) - F(p_L(y)) \geq F(x) - Q(p_L(y), y) : \tag{5.14}$$

Ahora bien, como f, g son convexas, tenemos

$$\begin{aligned} f(x) &\geq f(y) + \langle x - y, \nabla f(y)\rangle, \\ g(x) &\geq g(p_L(y)) + \langle x - p_L(y), \gamma(y)\rangle, \end{aligned}$$

donde $\gamma(y)$ está definida en el Lema 5.4 . Sumando entonces las inecuaciones de arriba, uno tiene lo siguiente:

$$F(x) \geq f(y) + \langle x - y, \nabla f(y)\rangle + g(p_L(Y)) + \langle x - p_L(y), \gamma(y)\rangle. \tag{5.15}$$

Por otro lado, por la definición de $p_L(y)$ se tiene que

$$Q(p_L(y), y) = f(y) + \langle p_L(y) - y, \nabla f(y)\rangle + \frac{L}{2}\|p_L(y) - y\|^2 + g(p_L(y)). \tag{5.16}$$

Por tanto, usando (5.15), (5.16) en (5.14) obtenemos

$$\begin{aligned} F(x) - F(p_L(y)) &\geq -\frac{L}{2}\|p_L(y) - y\|^2 + \langle x - p_L(y), \nabla f(y) + \gamma(y)\rangle \\ &= -\frac{L}{2}\|p_L(y) - y\|^2 + L\langle x - p_L(y), y - p_L(y)\rangle \\ &= \frac{L}{2}\|p_L(y) - y\|^2 + L\langle y - x, p_L(y) - y\rangle \end{aligned}$$

donde en la primera igualdad anterior usamos (5.10) □

Nota 5.7. *Señalemos que todos los resultados anteriores (y los próximos) son ciertos para cualquier espacio de Hilbert real H. Además, estos resultados pueden ser adaptados para el problema* (5.9) *con restricciones convexas. En tal caso, si $C \subset \mathbb{R}^n$ es un subconjunto cerrado, convexo y no vacío, el cálculo de p_L puede resultar complejo, a menos que C sea muy simple. Para simplificar, todos los resultados son desarrollados en un entorno sin restricciones y de dimensión finita.*

Podemos dar paso entonces a la descripción de nuestro método. Comenzaremos con la iteración básica de ISTA para resolver el problema (5.9).

1. **Entrada:** $L := L(f)$ (Constante de Lipschitz de ∇f)
2. Tomar $x_0 \in \mathbb{R}^n$.
3. **Update** ($k \geq 1$) Calcular
$$x_k = p_L(x_{k-1}) \tag{5.17}$$

Si $f(x) = \|Ax - b\|^2$ y $g(x) = \lambda\|x\|_1$ (con $\lambda > 0$), entonces el algoritmo anterior se reduce a un método iterativo básico de contracción (5.4) con $t = \frac{1}{L(f)}$. Este algoritmo resultará útil cuando p_L pueda ser calculado analíticamente o mediante un esquema de bajo coste. Esto suele ocurrir cuando g es separable, que en tal caso, el cálculo de p_L se reduce a resolver un problema de minimización de una dimensión.

Un posible inconveniente de este esquema es que la constante de Lipschitz $L(f)$ no siempre es conocida o se puede calcular. Por ejemplo, la constante de Lipschitz en el problema de regularización ℓ_1 (5.3) depende del máximo autovalor de A^tA. Para problemas de gran escala, esta cifra no es siempre fácilmente calculable. Por tanto, también analizaremos ISTA con una regla de tamaño de pasos hacia atrás:

1. Tomar $L_0 > 0$, $\eta > 1$ y $x_0 \in \mathbb{R}^n$.
2. ($k \geq 1$) Encontrar los enteros no negativos i_k tales que con $\bar{L} = \eta^{i_k} L_{k-1}$
$$F(p_{\bar{L}}(x_{k-1})) \leq Q_{\bar{L}}(p_{\bar{L}}(x_{k-1}), x_{k-1}). \tag{5.18}$$
Tomar $L_k = \eta^{i_k} L_{k-1}$ y calcular
$$x_k = p_{L_k}(x_{k-1}). \tag{5.19}$$

Nota 5.8. *Observemos que la sucesión de los valores de la función $(F(x_k))$ producido por ISTA es no creciente. Por supuesto, para cada $k \geq 1$,*

$$F(x_k) \leq Q_{L_k}(x_k, x_{k-1}) \leq Q_{L_{k-1}}(x_k, x_{k-1}) = F(x_{k-1}),$$

donde L_k es elegido por la regla de retroceso o $L_k \equiv L(f)$ es dado por la constante de Lipschitz de ∇f.

Nota 5.9. *Dado que la desigualdad (5.18) se satisface para $\bar{L} \geq L(f)$, donde $L(f)$ es la constante de Lipschitz de ∇f, se sigue que para ISTA con retroceso, uno tiene que $L_k \leq \eta L(f)$ para cada $k \geq 1$. En general,*

$$\beta L(f) \leq L_k \leq \alpha L(f), \tag{5.20}$$

donde $\alpha = \beta = 1$ para el ajuste de tamaño constante y $\alpha = \eta$, $\beta = \frac{L_0}{L(f)}$ para el caso hacia atrás.

5.1.3. Convergencia

Recordemos que ISTA se reduce al método del gradiente cuando $g(x) \equiv 0$. Para el método del gradiente, se sabe que la sucesión de los valores de la función $(F(x_k))$ converge a un valor óptimo de la función F_* a una velocidad de convergencia que no es peor que del orden de $O(\frac{1}{k})$, que también se denomina velocidad de convergencia "sublineal". Esto es, $F(x_k) - F_* \leq \frac{C}{k}$ para una constante positiva C.

Teorema 5.10. *Sea (x_k) la sucesión generada por (5.17) o (5.19). Entonces, para cualquier $k \geq 1$,*

$$F(x_k) - F(x^*) \leq \frac{\alpha L(f)\|x_0 - x^*\|^2}{2k} \quad \forall x^* \in X_*, \tag{5.21}$$

donde $\alpha = 1$ para el ajuste de tamaño de paso constante y $\alpha = \eta$ para el ajuste de tamaño de paso hacia atrás.

Demostración. Usando el Lema 5.6 con $x = x^*$,$y = x_n$ y $L = L_{n+1}$, obtenemos

$$\begin{aligned} \frac{2}{L_{n+1}}(F(x^*) - F(x_{n+1})) &\geq \|x_{n+1} - x_n\|^2 + 2\langle x_n - x^*, x_{n+1} - x_n\rangle \\ &= \|x^* - x_{n+1}\|^2 - \|x^* - x_n\|^2, \end{aligned}$$

que junto a (5.20) y el hecho de que $F(x^*) - F(x_{n+1}) \leq 0$ da lugar a

$$\frac{2}{\alpha L(f)}(F(x^*) - F(x_{n+1}) \geq \|x^* - x_{n+1}\|^2 - \|x^* - x_n\|^2. \tag{5.22}$$

Sumando esta inecuación sobre $n = 0, \ldots, k-1$ resulta

$$\frac{2}{\alpha L(f)}\left(kF(x^*) - \sum_{n=0}^{k-1} F(x_{n+1})\right) \geq \|x^* - x_k\|^2 - \|x^* - x_0\|^2. \tag{5.23}$$

Usando otra vez el Lema 5.6 con $x = y = x_n$ y $L = L_{n+1}$ se obtiene

$$\frac{2}{L_{n+1}}(F(x_n) - F(x_{n+1})) \geq \|x_n - x_{n+1}\|^2.$$

Como $L_{n+1} \geq \beta L(f)$, resulta que

$$\frac{2}{\beta L(f)}(F(x_n) - F(x_{n+1})) \geq \|x_n - x_{n+1}\|^2.$$

Multiplicando entonces esta última inecuación por n y sumando sobre $n = 0, \ldots, k-1$, conseguimos

$$\frac{2}{\beta L(f)} \sum_{n=0} k - 1(nF(x_n) - (n+1)F(x_{n+1}) + F(x_{n+1})) \geq \sum_{n=0}^{k-1} n\|x_n - x_{n+1}\|^2,$$

que se simplifica a

$$\frac{2}{\beta L(f)}\left(- kF(x_k) + \sum_{n=0}^{k-1} F(x_{n+1})\right) \geq \sum_{n=0}^{k-1} n\|x_n - x_{n+1}\|^2. \tag{5.24}$$

sumando (5.23) y (5.24) por β/α, obtenemos

$$\frac{2k}{\alpha L(f)}(F(x^*) - F(x_k)) \geq \|x^* - x_k\|^2 + \frac{\beta}{\alpha}\sum_{n=0}^{k-1} n\|x_n - x_{n+1}\|^2 - \|x^* - x_0\|^2,$$

y de ahí se sigue que

$$F(x_k) - F(x^*) \leq \frac{\alpha L(f)\|x - x_0\|^2}{2k}. \tag{5.25}$$

□

El resultado anterior se puede interpretar de la siguiente forma: el número de iteraciones de ISTA requerido para obtener un ε-solución óptima, es decir, un $\tilde{x}$ tal que $F(\tilde{x}) - F_* \leq \varepsilon$, es al menos $\lceil C/\varepsilon \rceil$, donde $C = \frac{\alpha L(f)\|x_0 - x^*\|^2}{2}$.

5.2. FISTA (Fast ISTA)

Como acabamos de ver, ISTA tiene un resultado de complejidad en el peor de los casos del orden de $O(1/k)$. Por tanto, vamos a tratar de dar un nuevo ISTA con un resultado de complejidad del orden de $O(1/k^2)$. Recordemos que cuando $g(x) \equiv 0$, el modelo general (5.7) consiste en minimizar un función convexa β-suave e ISTA lo reducía al método del gradiente. En este entorno, se puede demostrar que existe un método del gradiente con un resultado de complejidad de $O(1/k^2)$, el cual es un método de primer orden óptimo para problemas suaves, en el sentido de Nemirovsky y Yudin [**11**]. El extraordinario hecho es que el método desarrolado en Y. E. Nesterov [**12**] no requiere más que una evaluación del gradiente en cada iteración (es decir, igual que el método del gradiente) sino sólo un punto adicional que se elige inteligentemente, y es fácil de calcular.

Por tanto, extenderemos el método de Y. E. Nesterov [**12**] al modelo general (5.7) y estableceremos el resultado de complejidad mejorado. Nuestro análisis también proporciona una prueba simple para el caso especial suave (es decir, $g \equiv 0$). Comenzaremos dando el algoritmo para un tamaño de paso constante.

1. **Entrada:** $L = L(f)$ (constante de Lipschitz de ∇f).
2. Tomar $y_1 = x_0 \in \mathbb{R}^n$, $t_1 = 1$.
3. Para $k \geq 1$,

$$x_k = p_L(y_k), \tag{5.26}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4_k^2}}{2}, \tag{5.27}$$

$$y_{k+1} = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1}). \tag{5.28}$$

La principal diferencia entre FISTA e ISTA es que el operador iterativo de encogimiento p_L no es empleado en el punto anterior x_{k-1}, sino en el punto y_k, el cual usa una combinación lineal específica de los dos puntos anteriores $\{x_{k-1}, x_{k-2}\}$. Bien es cierto que el principal esfuerzo (computacional) tanto en ISTA como en FISTA sigue siendo el mismo, es decir, en el operador p_L. El cálculo adicional requerido por FISTA en los pasos (5.27) y (5.28) es claramente marginal. Por otro lado, la fórmula (5.27) surge de la relación recursiva que se verá a continuación en el próximo lema. Pero también analizaremos FISTA con una regla de longitud de paso hacia atrás, que vamos a dar explícitamente.

1. Tomar $L_0 > 0$, $\eta > 1$, y $x_0 \in \mathbb{R}^n$. Sea $y_1 = x_0$, $t_1 = 1$.

2. Para $k \geq 1$, encontrar el menor entero no negativo i_k tal que, con $\bar{L} = \eta^{i_k} L_{k-1}$

$$F(p_{\bar{L}}(y_k)) \leq Q_{\bar{L}}(p_{\bar{L}}(y_k), y_k).$$

Sea $L_k = \eta^{i_k} L_{k-1}$. Calcular

$$\begin{aligned} x_k &= p_{L_k}(y_k), \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ y_{k+1} &= x_k + \left(\frac{t_k - 1}{t_{k-1}}\right)(x_k - x_{k-1}). \end{aligned}$$

Además, se tiene que las cotas superiores e inferiores vistas en la Nota 5.9 siguen siendo ciertas para FISTA, es decir,

$$\beta L(f) \leq L_k \leq \alpha L(f).$$

El siguiente resultado nos da una relación recursiva fundamental para la sucesión $(F(x_k) - F(x^*))$ que implicará una mejor velocidad de complejidad del orden de $O(1/k^2)$.

Lema 5.11. *Las sucesiones generadas por FISTA (x_k, y_k) con una constante cualquiera o una regla de longitud de paso hacia atrás satisfacen que, para $k \geq 1$,*

$$\frac{2}{L_k} t_k^2 v_k - \frac{2}{L_{k+1}} t_{k+1}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2,$$

donde $v_k := F(x_k) - f(x^*)$, $u_k := t_k x_k - (t_k - 1)x_{k-1} - x^*$.

Demostración. Comenzamos aplicando el Lema 5.6 en los puntos $x := x_k$,$y := y_{k+1}$ con $L = L_{k+1}$ e igualmente en los puntos $x := x^*$,$y := y_{k+1}$, obtenemos lo siguiente

$$\begin{aligned} 2L_{k+1}^{-1}(v_k - v_{k+1}) &\geq \|x_{k+1} - y_{k+1}\|^2 + 2\langle x_{k+1} - y_{k+1}, y_{k+1} - x_k\rangle, \\ 2L_{k+1}^{-1} v_{k+1} &\geq \|x_{k+1} - y_{k+1}\|^2 + 2\langle x_{k+1} - y_{k+1}, y_{k+1} - x^*\rangle, \end{aligned}$$

donde hemos usado el hecho de que $x_{k+1} = p_L(y_{k+1})$. Para conseguir una relación entre v_k y v_{k+1} multiplicaremos la primera de las inecuaciones anteriores por $(t_{k+1} - 1)$ y se lo sumaremos a la segunda de ellas:

$$\begin{aligned} \frac{2}{L_{k+1}}((t_{k+1} - 1)v_k - t_{k+1} v_{k+1}) &\geq t_{k+1}\|x_{k+1} - y_{k+1}\|^2 + \\ &+ 2\langle x_{k+1} - y_{k+1}, t_{k+1} y_{k+1} - (t_{k+1} - 1)x_k - x^*\rangle. \end{aligned}$$

Multiplicando esta última inecuación por t_{k+1} y usando la relación $t_k^2 = t_{k+1}^2 - t_{k+1}$ que se tiene gracias a (5.26), obtenemos

$$\frac{2}{L_{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}(x_{k+1} - y_{k+1})\|^2 + \\ +2t_{k+1}\langle x_{k+1} - y_{k+1}, t_{k+1}y_{k+1} - (t_{k+1} - 1)x_k - x^*\rangle.$$

Aplicando la relación de Pitágoras

$$\|b - a\|^2 + 2\langle b - a, a - c\rangle = \|b - c\|^2 - \|a - c\|^2$$

al lado derecho de la última inecuación con

$$a := t_{k+1}y_{k+1}, \quad b := t_{k+1}x_{k+1}, \quad c := (t_{k+1} - 1)x_k + x^*$$

obtenemos así

$$\frac{2}{L_{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}x_{k+1} - (t_{k+1} - 1)x_k - x^*\|^2 - \|t_{k+1}y_{k+1} - (t_{k+1} - 1)x_k - x^*\|^2.$$

Por tanto, con y_{k+1} y u_k definidos como

$$t_{k+1}y_{k+1} = t_{k+1}x_k + (t_k - 1)(x_k - x_{k+1}), \quad u_k = t_k x_k - (t_k - 1)x_{k-1} - x^*,$$

se sigue que

$$\frac{2}{L_{k+1}}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|u_{k+1}\|^2 - \|u_k\|^2,$$

que junto a que $L_{k+1} \geq L_k$ conseguimos

$$\frac{2}{L_k}t_k^2 v_k - \frac{2}{L_{k+1}}t_{k+1}^2 v_{k+1} \geq \|u_{k+1}\|^2 - \|u_k\|^2.$$

□

El número de iteraciones de FISTA requerido para obtener una solución ε-óptima es decir, un $\tilde{x}$ tal que $F(\tilde{x}) - F_* \leq \varepsilon$, es como máximo $\lceil C/\sqrt{\varepsilon} - 1 \rceil$, donde $C = \sqrt{2\alpha L(f)\|x_0 - x^*\|^2}$, y que obviamente mejora ISTA.

Ejemplo 5.12. *Veamos una comparación de cómo actúa FISTA respecto al básico ISTA. Para ello, vamos a considerar problemas mal condicionados (el autovalor más pequeño de A^tA es próximo a cero, y el máximo es 1).*

Ambos métodos con una regla de longitud de paso constante y se aplicaron al problema ℓ_1 de regularización, que es, $f(x) = \|Ax - b\|^2$ y $g(x) = \lambda\|x\|_1$. En todas las simulaciones, observamos que FISTA superó significativamente ISTA con respecto al número de iteraciones requeridas para conseguir una precisión dada.

Todos los píxeles de las imágenes originales descritos en los ejemplos están en una escala de rango entre 0 y 1. La imagen pasó por un desenfoque Gaussiano de 9×9 y una desviación estándar 4 seguido de una perturbación Gaussiana blanca de media adicional 0 y desviación estándar de 10^{-3}. Tanto la imagen original como la observada se puede ver en la siguiente imagen. Para estos experimentos vamos a asumir condiciones de fronteras reflexivas (Neumann). Luego probamos ISTA y FISTA para resolver el problema (5.3), donde b representa la imagen observada, y $A = RW$, donde R es la matriz representativa del operador de desenfoque y W es la inversa de una transformación de ondas de Haar de tres etapas. El parámetro de regularización fue escogido como $\lambda = 2e - 5$, y la imagen inicial fue la imagen desenfocada. La constante de Lipschitz fue calculada, ya que los autovalores de la matriz A^tA pueden ser calculados fácilmente usando la transformación 2-dimensional del coseno. Las iteraciones 100 y 200 son descritas en la Figura 5.2 . El valor de la función en la iteración k es denotado como F_k. Las imágenes producidas por FISTA son de mejor calidad que estas mismas creadas por ISTA.

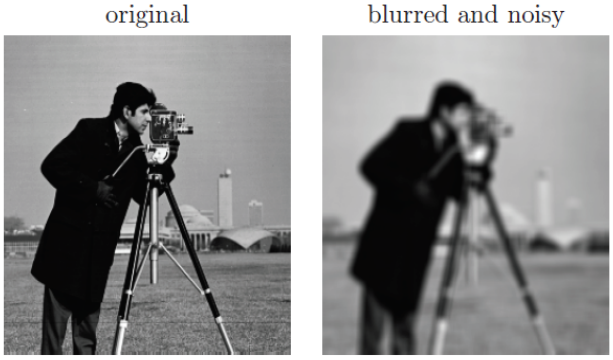

Figura 5.1: Desvanecimiento del fotógrafo

Figura 5.2: Iteraciones con los distintos métodos

Por otro lado, el valor de la función de FISTA fue consistentemente más bajo que los valores de la función de ISTA; también calculamos los valores de la función producidos después de 1000 *iteraciones de ISTA y FISTA, los cuales, fueron respectivamente* $2{,}45e-1$ *y* $2{,}23e-1$*. Por tanto, el valor de la función de ISTA después de* 1000 *iteraciones es todavía peor (de hecho, mucho más) que el valor de la función de FISTA después de* 100 *iteraciones.*

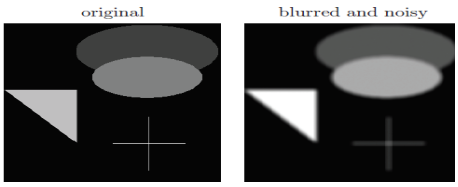

Figura 5.3: Desenfoque de la imagen prueba

Capítulo 6

El problema LASSO

> "I never keep a scorecard or the batting averages. I hate statistics. What I got to know, I keep in my head."

Esta es una cita del lanzador de béisbol Dizzy Dean, que jugó en las mejores ligas desde 1930 hasta 1947. ¡Cómo ha cambiado el mundo en estos últimos 74 años! Bien es cierto que ahora las grandes cantidades de datos se recolectan y extraen en cada área de la ciencia, entretenimiento, negocios e industria. Es decir, el mundo actual está inundado de datos. Pero como Rutherford D. Rogers (y otros tanto) dijo:

> "We are drowning in information and starving for knowledge."

Hay una necesidad crucial de clasificar esta masa de información y reducirla a lo esencial. Para que este proceso sea exitoso, tenemos que esperar que el mundo no sea tan complejo como podría ser. Por ejemplo, esperamos que ninguno de los 30000 genes del cuerpo humano no estén directamente involucrados en el proceso que conduce al desarrollo del cáncer. Esto apunta a una suposición subyacente de simplicidad.

En los capítulos anteriores hemos descritos diferentes métodos para la resolución de un problema particular: el problema LASSO, que no es más que un modelo de selección de variables, que penaliza el valor de los coeficientes β_j, para no tener así un modelo con demasiadas variables ni con un sobreajuste. Sin embargo, en ningún momento hemos dado justificación alguna de por qué este problema adquiere tal relevancia. Esto es debido a que el problema LASSO deriva de la regresión lineal, uno de los grandes conocidos en el mundo de las matemáticas, para el problema de mínimos cuadrados.

6.1. Introducción

El ejemplo líder es la regresión lineal, en la cual, tenemos N observaciones de una variable de salida y_i y p variables de predicción asociadas $x_i = (x_{i1}, \cdots, x_{ip})^t$. El objetivo no es más que obtener el valor de las predicciones, ya sea la predicción real con datos futuros como saber qué predicciones juegan un papel importante. Un modelo lineal de regresión asume que

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + e_i, \tag{6.1}$$

donde $\beta_0 = (\beta_1, \beta_2, \ldots, \beta_p)$ son parámetros conocidos y e_i es un término de error. El método de mínimos cuadrados proporciona estimaciones de los parámetros a través de la función de mínimos cuadrados

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2. \tag{6.2}$$

Normalmente, todas las estimaciones de mínimos cuadrados de (6.2) serán distintas de cero. Ésto hará que la interpretación final del modelo resulte desafiante si p es muy grande. De hecho, si $p > N$, estas estimaciones no serán únicas. Por tanto, hay una necesidad de construir o regularizar el proceso de estimación. En LASSO o la regresión regularizada ℓ_1, estimamos los parámetros resolviendo el siguiente problema:

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \quad sujeto\ a \quad \|\beta\|_1 \leq t \tag{6.3}$$

Podemos pensar sobre t como una estimación de la norma total de ℓ_1 del vector de parámetro y lasso encuentra el mejor ajuste dentro de esta estimación. Si la estimación t es lo suficientemente pequeña, LASSO produce vectores de solución escasos, con sólo algunas coordenadas distintas de cero, cosa que no ocurre para la norma de ℓ_q con $q > 1$; sin embargo, para $q < 1$, las soluciones son escasas, pero el problema no es convexo y esto hace que la minimización resulte un reto computacional. En gran medida, la convexidad simplifica los cálculos al igual que el supuesto de dispersión en sí. Permite a ciertos algoritmos que puedan manejar incluso problemas con millones de parámetros.

Por lo tanto, las ventajas de escasez son una interpretación del modelo ajustado y la conveniencia computacional. Pero en los últimos años, apareció una nueva ventaja desde el análisis matemático profundo de este área. Este término ha sido la apuesta por el "principio de escasez":

> "Use a procedure that does well in sparse problems, since no procedure does well in dense problems."

Podemos pensar en esto en términos de la cantidad de información de N/p por parámetro. Si $p \gg N$ y el verdadero modelo no es escaso, entonces el número de muestras N es demasiado pequeño para permitir una estimación precisa de los parámetros. Pero sin embargo, si es escaso, de modo que sólo $k < N$ parámetros son realmente distintos de cero en el verdadero modelo subyacente, entonces resulta que podemos estimar los parámetros de manera efectiva, usando LASSO y otros métodos relatados en [**7**]. Esto puede resultar algo sorprendente, porque somos capaces de hacer esto incluso aunque no se nos diga qué k de los p parámetros son distintos de cero. Obviamente, no podríamos hacerlo tan bien como podríamos si tuviéramos esa información, pero resulta que podemos hacerlo aún razonablemente bien.

En la regresión lineal, se dan N muestras $\{(x_i, y_i)\}_{i=1}^{N}$, donde cada $x_i = (x_{i1}, \ldots, x_{ip})$ es un vector futuro (o de predicciones) p-dimensional, y cada $y_i \in \mathbb{R}$ es la variable de respuesta asociada. Nuestro objetivo va a ser aproximar la variable respuesta y_i

usando una combinación lineal de predicciones

$$\eta(x_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \tag{6.4}$$

El modelo es parametrizado por un vector de regresión de pesos $\beta = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$ un término $\beta_0 \in \mathbb{R}$. El estimador usual de "mínimos cuadrados" para el par (β_0, β) está basado en minimizar la pérdida del error cuadrático:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \right\}. \tag{6.5}$$

Hay dos razones por las que podríamos considerar una alternativa para la estimación de mínimos cuadrados. La primera de ellas es la *precisión de la predicción*: la estimación de mínimos cuadrados con frecuencia tiene un sesgo bajo pero gran varianza, y la precisión de la predicción a veces puede ser mejorada por encogimiento de los valores de los coeficientes de regresión, o estableciendo algunos coeficientes iguales a cero. Haciendo esto, introducimos algunos sesgos pero reducimos la varianza de los valores predecidos, y por tanto, podrían mejorar la precisión general de las predicciones. La segunda razón es para fines de interpretación. Con un gran número de predictores, a menudo nos gustaría identificar un subconjuntos más pequeño de éstos que exhiben los efectos más fuertes.

En relación con la solución de mínimos cuadrados, esta restricción tiene el efecto de reducir los coeficientes e incluso establecer algunos como cero. De esta manera, resulta una manera automática para realizar la selección del modelo en la regresión lineal. Además, a diferencia de otros criterios para la selección de modelos, el problema de optimización resultante es convexo por lo que puede ser resuelto de una manera eficiente para problemas de gran escala.

6.1.1. LASSO

El método LASSO (*Least Absolute Shrinkage and Selection Operator*), que no es más que la regresión generalizada en ℓ_1, trata de penalizar un tamaño grande de β en el problema de mínimos cuadrados para la regresión lineal. Aunque a simple vista parece una desventaja, pues ahora no tenemos una función diferenciable en $\mathbb{R}^n$, por otro lado, nos da una mayor interpretabilidad del problema, al proporcionar en el caso $p \gg N$ soluciones con un alto número de variables denominado "selección de variables".

Dada una colección de N pares de predicciones-respuestas $\{(x_i, y_i)\}_{i=1}^{N}$, LASSO encuentro la solución $(\widehat{\beta}_0, \widehat{\beta})$ para el problema de optimización (6.3), que es equivalente al siguiente:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2N} \|y - \beta_0 - X\beta\|_2^2 \right\} \quad sujeto \quad \|\beta\|_1 \leq t. \tag{6.6}$$

La cota t limita la suma de los valores absolutos de las estimaciones de los parámetros. Dado que una estimación de parámetro reducida corresponde a un modelo más restringido, t limita lo bien que podemos ajustar los datos. Primero normalizamos los predictores X para que cada columna esté centrada y tenga varianza unitaria

(esto es $\frac{1}{N}\sum_{i=1}^{N} x_{ij} = 0$ y $\frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1$). Por conveniencia, también asumiremos que los valores de salida y_i habrán sido centrados (es decir, $\frac{1}{N}\sum_{i=1}^{N} y_i = 0$). Estas hipótesis servirán de cara a que podremos omitir el término β_0 en la optimización LASSO. Dada una solución óptima β centrada en los datos, podremos recuperar las soluciones óptimas para los datos no centrados; $\widehat{\beta}$ es el mismo, y β_0 viene dado por

$$\widehat{\beta_0} = \bar{y} - \sum_{j=1}^{p} \widehat{\beta_j}\bar{x}_j.$$

Con frecuencia, el problema LASSO es reescrito de la forma Lagrangiana, que es:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \right\}, \tag{6.7}$$

para algún $\lambda \geq 0$. Gracias a la dualidad lagrangiana, hay una correspondencia uno a uno entre el problema con restricción (6.6) y la forma Lagrangiana (6.7): es decir, para cada valor de t en rango donde la restricción $\|\beta\|_1 \leq t$ es activa, hay un valor correspondiente de λ que produce la misma solución de la forma Lagrangiana. De forma inversa, la solución $\widehat{\beta}_\lambda$ del problema (6.7) resuelve el problema acotado con $t = \|\widehat{\beta}_\lambda\|_1$. En muchas descripciones de LASSO, el factor $1/2N$ que aparece en (6.6) y (6.7) es reemplazado por $1/2N$ ó 1. Aunque esto no hace ninguna diferencia en (6.6), y corresponde a una reparametrización de λ en (6.7), este tipo de normalización hace que los valores de λ sean comparables para diferentes tamaños de muestra.

La teoría de análisis convexo nos dice que las condiciones necesarias y suficientes para una solución del problema (6.7) tiene la forma

$$-\frac{1}{N}\langle x_j, y - X\beta\rangle + \lambda s_j = 0\ ,\ j = 1, ..., p. \tag{6.8}$$

Aquí cada s_j es una cantidad desconocida igual a $sg(\beta_j)$ si $\beta_j \neq 0$ y algún valor entre $[-1, 1]$, de lo contrario, es un subgradiente para la función valor absoluto. Es decir, las soluciones $\widehat{\beta}$ para el problema (6.7) son las mismas soluciones que $(\widehat{\beta}, \widehat{s})$ de (6.8). Este sistema es una forma de las llamadas **Karush-Kuhn-Tucker(KKY)** condiciones para el problema (6.7). La cota t en criterio de LASSO controla la complejidad del modelo; grandes valores de t "liberan" más parámetros, permitiendo así, permitir al modelo acercarse más a los datos. De forma inversa, pequeños valores de t restringen los parámetros más, lo que conduce a modelos más dispersos y más interpretables que se ajustan menos a los datos. Olvidemos por un momento la interpretabilidad, podemos preguntarnos por el valor de t que nos da el modelo más preciso para predecir datos de prueba independientes de la misma población. Tal precisión es conocida como la *capacidad de generalización del modelo*. Un valor de t demasiado pequeño puede evitar que LASSO capture la señal principal en los datos, mientras que un valor demasiado grande puede provocar un sobreajuste. En ambos casos, aumentará el error de predicción en un conjunto de prueba.

Sí es verdad que suele haber un valor intermedio de t que logra un equilibrio entre estos dos extremos, y en el proceso, produce un modelo con algunos coeficientes iguales a cero. Para estimar este mejor valor de t, podemos crear un "entrenamiento artificial" y conjuntos de prueba dividiendo el conjunto de datos dado al azar y estimando el rendimiento de los datos de prueba, utilizando un procedimiento conocido como *validación cruzada*.

Para más detalle, en primer lugar, dividimos de manera aleatoria el conjunto completo de datos en un número de grupos $K > 1$. Las elecciones de K serían 5 ó 10, y a veces N. Tomamos un grupo como *muestra test*, y el resto de $K-1$ grupos como *muestra de aprendizaje*. Posteriormente, aplicamos LASSO a la muestra de aprendizaje para un rango de diferentes valores de t, y usamos cada modelo ajustado para predecir las respuestas en la muestra test, registrando los errores de predicción cuadrático medio para cada valor de t. Este proceso es repetido un total de K veces, con cada uno de los K grupos dándoles la oportunidad de jugar el papel de muestra test y el resto d elos $K-1$ grupos como muestra de aprendizaje. De esta forma, obtenemos K estimadores diferentes del error de predicción sobre un rango de valores de t. Estos K estimadores son promediados para cada valor de t, produciendo así una *curva de error de validación cruzada*.

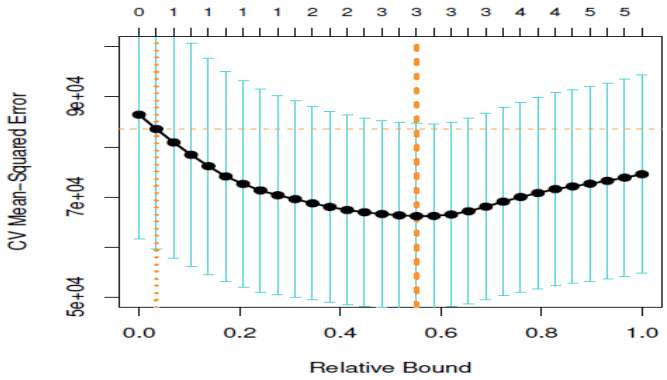

Figura 6.1: Estimación con validación cruzada del error de predicción cuadrático medio, como una función de la cota relativa en ℓ_1 de $\widetilde{t} = \|\widehat{\beta}(t)\|_1 / \|\widetilde{\beta}\|_1$, donde $\widehat{\beta}(t)$ es el estimador LASSO correspondiente a la cota t en ℓ_1 y $\widetilde{\beta}$ es la solución de mínimos cuadrados.

La Figura 6.1 muestra esta curva para el ejemplo de "crime-data", obtenido usando $K = 10$ particiones. Dibujamos la estimación de la predicción del error cuadrático medio frente a la cota relativa $\widetilde{t} = \|\widehat{\beta}(t)\|_1 / \|\widetilde{\beta}\|_1$, donde el estimador $\widehat{\beta}(t)$ corresponde a la solución de LASSO para la cota t y β es la solución general de mínimos cuadrados. Las barras de error que aparecen en la figura anterior indican más o menos un error estándar en las estimaciones con validación cruzada del error de predicción. Una línea de puntos vertical es dibujada en la posición del mínimo, esto es $\widetilde{t} = 0{,}56$, mientras que una línea punteada es dibujada con la elección de una "regla de un error estándar", que es $\widetilde{t} = 0{,}003$. El número de coeficientes distintos de cero en cada modelo se muestra en la parte superior. Hemos de tener en cuenta que el proceso de validación cruzada de arriba se centra en párametro vinculado t. También se puede realizar una validación cruzada en la forma Lagrangiana 6.7 centrándose en el párametro λ. Ambos métodos darán resultados similares pero no idénticos, ya que el "mapeo" entre t y λ depende de los datos.

6.1.2. Regresión logística

Con una respuesta binaria codificada de la forma $Y \in \{0,1\}$, el **modelo lineal logístico** es usado con frecuencia: modela la "log-probabilidad" radio como una combinación lineal

$$\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right) = \beta_0 + \beta^t x, \tag{6.9}$$

donde $X = (X_1, X_2, \ldots, X_p)$ es un vector de predicciones, $\beta_0 \in \mathbb{R}$ y $\beta \in \mathbb{R}^p$ es el vector de coeficientes de regresión. Invirtiendo esta transformación resulta una expresión para la probabilidad condicionada

$$P(Y=1|X=x) = \frac{e^{\beta_0+\beta^t}}{1+e^{\beta_0+\beta^t x}}. \tag{6.10}$$

Sin ninguna restricción en los parámetros (β_0, β), el modelo especifica la probabilidad que ese encuentra entre $(0,1)$. Normalmente, ajustamos los modelos logísticos maximizando la probabilidad logarítmica binomial de los datos.

La transformación *logit* de las probabilidades condicionales es un ejemplo de "link function". En general, una "link function" es una transformación de la media condicional $\mathbb{E}[Y|X=x]$, de manera que, en este caso, la probabilidad condicionada que $Y=1$ a una escala más natural en la que los parámetros puede ajustarse sin restricciones. Otro ejemplo, si la variable Y representa cuentas, tomando valores en $\mathbb{N}$, entonces necesitamos asegurarnos de que la media condicional es positiva. Una elección natural es el modelo logarítmico lineal

$$\log(\mathbb{E}[Y|X=x]) = \beta_0 + \beta^t x, \tag{6.11}$$

con su "log link function". Aquí ajustamos los parámetros maximizando la probabilidad logarítmica de Poisson de los datos. Los modelos (6.9) y (6.11) describen la variable respuesta usando un miembro de la *familia exponencial*, los cuales incluye Bernouilli, Poisson y Gaussiano como casos particulares. Una versión transformada de la media de la respuesta $\mathbb{E}[Y|X=x]$ es entonces aproximada por un modelo lineal. De hecho, si usamos $\mu(x) = \mathbb{E}[Y|X=x]$ para denotar la media condicionada de Y dado $X=x$, entonces **GLM (General Lineal Model)** está basado en un modelo de la forma

$$g[\mu(x)] = \underbrace{\beta_0 + \beta^t x}_{\eta(x)}, \tag{6.12}$$

donde $g : \mathbb{R} \to \mathbb{R}$ es una link function estrictamente monótona.

Los modelos lineales generalizados también pueden ser usados para modelar las respuestas de varias categorías como ocurre en muchos problemas, incluyendo clasificación de dígitos escritos a manos o reconocimiento de voz. La multinomial reemplaza la distribución binomial, y para ello usamos una representación lineal logarítmica:

$$P[Y=y|X=x] = \frac{e^{\beta_{0k}+\beta_k^t x}}{\sum_{l=1}^K e^{\beta_{0l}+\beta_l^t x}} \tag{6.13}$$

de manera que hay K coeficientes para cada variable.

La regresión logística ha sido popular desde hace medio siglo en la investigación biomédica, y recientemente, ha ganado popularidad para modelar una gama más alta

de datos. En el ámbito de grandes dimensiones, en el que el número de p es mayor que el de la muestra, no se puede utilizar sin modificaciones. Cuando $p > N$, cualquier modelo lineal está "sobreparametrizado", y se necesita regularización para lograr un ajuste estable. Estos modelos de gran dimensión se adaptan a diversas aplicaciones. Por ejemplo, los problemas de clasificación de documentos pueden involucrar características binarias sobre un diccionario predefinido $p = 20000$ o más palabras. Cuando la característica es binaria, normalmente codificada como 0 ó 1. Entonces, la atención se enfoca en estimar la probabilidad condicionada $P(Y = 1|X = x) = \mathbb{E}[Y|X = x]$. Dado el modelo logístico (6.9), la probabilidad logarítmica negativa con la regularización ℓ_1 toma la forma

$$-\frac{1}{N}\sum_{i=1}^{N}\{y_i \log(P(Y=1|x_i) + (1-y_i)\log(P(Y=0|x_i)\} + \lambda\|\beta\|_1$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\left\{y_i(\beta_0 + \beta^t x_i) - \log(1 + e^{\beta_0+\beta^t x_i})\right\} + \lambda\|\beta\|_1. \tag{6.14}$$

En la comunidad del machine learning, es más común codificar la variable respuesta Y en términos de los signos de las variables $\{-1, +1\}$ en lugar de $\{0, 1\}$; cuando usemos el signo de las variables, la probabilidad logarítmica penalizada tiene la forma

$$\frac{1}{N}\sum_{i=1}^{N}\log(1 + e^{-y_i f(x_i;\beta_0,\beta)}) + \lambda\|\beta\|_1, \tag{6.15}$$

donde $f(x_i; \beta_0, \beta) := \beta_0 + \beta^t x_i$. Para un par covariable-respuesta (x, y), el producto de $y \cdot f(x)$ se denomina *margen* (*margin* en inglés): un margen positivo significa una clasificación correcta, mientras que un margen negativo significa todo lo contrario. De la forma de la probabilidad logarítmica de (6.15), vemos que maximizando la probabilidad equivale a minimizar una función de pérdida monótona decreciente en los márgenes.

6.2. Descenso por coordenadas

Cierta clase de problemas, entre ellos LASSO y sus variantes, tiene una propiedad de separabilidad adicional que se debe a un algoritmo de minimización de coordenadas. El **descenso por coordenadas** es un algoritmo iterativo que se actualiza desde βt hasta β^{t+1} eligiendo una sola coordenada para actualizar, para luego realizar una minimización univariante sobre esta coordenada. En concreto: si la coordenada k es elegida en la iteración t, entonces la actualización viene dada por

$$\beta_k^{t+1} = \text{argmin}_{\beta_k} f(\beta_1^t, \beta_2^t, ..., \beta_{k-1}^t, \beta_k^t, \beta_{k+1}^t, ..., \beta_p^t), \tag{6.16}$$

y $\beta_j^{t+1} = \beta_k^t$ para $j \neq k$. Una elección muy común sería recorrer las coordenadas en un orden fijo. Este enfoque también se puede generalizar para descenso por coordenadas, en el que las variables se dividen en bloques que no se superponen, y realizamos la minimización en un solo bloque en cada ronda. Una pregunta normal

sería cuándo este procedimiento converge al mínimo global de una función convexa: una condición suficiente (pero algo restrictiva) es que f sea continuamente diferenciable y y estrictamente convexa en cada coordenada. Sin embargo, el uso de varios reguladores estadísticos nos lleva a problemas de optimización que no necesitan ser diferenciables. Para tales casos, se requiere más cuidado al usar la minimización de coordenadas porque como veremos a continuación, puede atascarse en puntos que no sean óptimos. Una estructura del problema que asegura un buen comportamiento de la minimización de coordenadas es un tipo de condición de separabilidad. En concreto, supongamos que función coste f tiene descomposición aditiva, esto es

$$f(\beta_1, ..., \beta_p) = g(\beta_1, ..., \beta_p) + \sum_{j=1}^{p} h_j(\beta_j) \tag{6.17}$$

donde $g : \mathbb{R}^p \to \mathbb{R}$ es convexa y diferenciable, y las funciones univariantes $h_j : \mathbb{R} \to \mathbb{R}$ son convexas (y no necesariamente diferenciables). [**15**] y [**16**] muestra que para cualquier función coste f convexa con la estructura separable anterior está garantiza su convergencia al mínimo global. La propiedad clave que subyace a este resultado es la separabilidad de la componente no diferenciable $h(\beta) = \sum_{j=1}^{p} h_j(\beta_j)$, como una suma de funciones de cada parámetro. Este resultado implica que el descenso por coordenadas es un algoritmo adecuado para LASSO así como para otros problemas. Pero cuando la componente no diferenciable h no es separable, el descenso por coordenadas ya no se garantiza que converja; en cambio, es posible crear problemas, para los cuales, se atascará y fallarán a la hora de alcanzar el mínimo global.

Ejemplo 6.1. *Como una ilustración, consideremos una instancia de un problema que incumple (6.17); aquí la componente no diferenciable toma la forma* $h(\beta) = \sum_{j=1}^{p} h_j(\beta_j)$. *La figura que viene a continuación nos muestra la dificultad.*

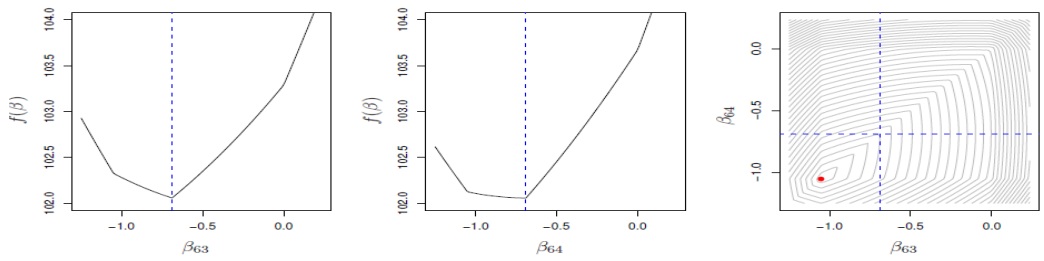

Figura 6.2: Fallo del descenso de coordendas en un problema de "fused LASSO" (LASSO fusionado) con 100 parámetros. Los valores óptimos para dos de los valores, β_{63} y β_{64}, son ambos $-1{,}05$, como se muestra con el punto en la imagen derecha. Las imágenes de la izquierda y del centro muestran cortes de la función objetivo f como una función de β_{63} y β_{64} con los otros parámetros establecidos en los mínimos globales. El mínimo de coordenadas,tanto en β_{63} como en β_{64} es $-0{,}69$ en lugar de $-1{,}05$. La imagen derecha muestra los contornos de la superficie de dos dimensiones. El algoritmo se atasca en el punto$(-0{,}69, -0{,}69)$. Debido a la convexidad estricta, la superficie tiene esquinas en las cuales, el algoritmo puede quedarse atascado. Con el fin de llegar al mínimo, tenemos que mover tanto β_{63} como β_{64} juntos.

Hemos creado un problema de LASSO fusionado con 100 parámetros, con las soluciones de dos parámetros, $\beta_{63} = \beta_{64} \approx -1$. Se puede ver que el algoritmo de descenso por coordenadas se ha quedado atascado en una esquina de la superficie de respuesta, y está estacionario bajo movimientos de una sola coordenada.

Tseng [**16**] da una condición de convergencia más general e intuitiva del descenso por coordenadas, una que depende del comportamiento the de la derivada direccional de la función coste f. Para una dirección dada $\Delta \in \mathbb{R}^p$, la derivada direccional inferior viene dada por:

$$f'(\beta; \Delta) := \liminf_{s \downarrow 0} \frac{f(\beta + s\Delta) - f(\beta)}{s}. \tag{6.18}$$

Aproximadamente, un algoritmo de descenso por coordenadas sólo gana información sobre la dirección de la forma $e^j = (0, 0, ..., 0, e_j, 0, ..., 0)$ para algún $e_j \in \mathbb{R}$. Por lo tanto, supongamos que el algoritmo alcanza un punto β para el cual

$$f'(\beta; e^j) \geq 0, \; \forall j \; = 1, ..., p, \text{ y vectores coordenados } e^j. \tag{6.19}$$

En cualquier punto, no hay direcciones de coordenadas que reduzcan aún más el valor de la función. Por lo tanto, necesitamos cualquier β que satisfaga la condición (6.19) y también $f'(\beta; \Delta) \geq 0$ para cualquiera que sea la dirección $\Delta \in \mathbb{R}^p$. Tseng [**16**] lo llama la *condición de regularidad*. Descarta una situación como la figura vista en el ejemplo, en la que los movimientos a lo largo de todas las direcciones de coordenadas no disminuyen el criterio.

Nota 6.2. *Vale la pena observar que la separabilidad de la componente no diferenciable de la función objetivo implica regularidad, pero hay funciones ni diferenciables ni separables que aún son regulares. Un ejemplo de ello es la función*

$$h(\beta_1, ..., \beta_p) = |\beta|^t \boldsymbol{P} |\beta| = \sum_{j,k=1}^{p} |\beta_j| P_{jk} |\beta_k|,$$

donde $\boldsymbol{P}$ es una matriz simétrica definida positiva.

6.2.1. Comparación entre el descenso de Nesterov y el descenso por coordenadas

Tanto el algoritmo de descenso por coordenadas como el método del gradiente de Nesterov son enfoques simples y computacionalmente eficientes para resolver el problema LASSO. Por tanto, nos preguntamos cómo podemos compararlos en términos de costes computacionales por cada iteración. Si el iterante β^t tiene k coeficientes distintos de cero, cada paso de descenso de coordenadas, sobre todos los p predictores toma un número de operaciones del orden $O(pN + kN)$. Por otro lado, la actualización del gradiente generalizado, que es $\beta^{t+1} = \mathcal{S}_{s^t\lambda}\Big(\beta^t + s^t \frac{1}{N} \mathbf{X}^t(\mathbf{y} - \mathbf{X}\beta^t)\Big)$, requiere unas operaciones del orden $O(kN)$ para calcular el producto de la matriz-vector $\mathbf{X}\beta$, y calcular el producto $\mathbf{X}^t(\mathbf{y} - \mathbf{X}\beta)$, con otra vez, un número total de operaciones del orden de $O(pN + kN)$.

Con el fin de examinar de forma más exhaustiva la eficiencia relativa del descenso por coordenadas, y el de Nesterov, realizamos un pequeño estudio de simulación. Para

ello, generamos una matriz de predicción $\mathbf{X}$ de orden $N \times p$ con entradas gaussianas normalizadas y correlación por pares 0 ó 0,5 entre características. Los coeficientes β_j fueron definidos como $|\beta_j| = \exp[-\frac{1}{2}(u(j-1))^2]$ con $u = \sqrt{\pi/20}$ y signos alternativos $+1, -1, +1, ...$ La salida y_i fue generado como

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \sigma\varepsilon_i \tag{6.20}$$

con σ elegido como esa relación señal-ruido $Sd[E(y_i)]\sigma$ igual a 3. La tabla siguiente muestra el promedio de tiempos de una CPU para el descenso por coordenadas, para un escenario con $N > p$ y otro con $N < p$. Se utilizaron iniciaciones suaves en cada caso, con la convergencia definida como el cambio máximo en el vector de parámetros inferior a 10^{-4}.

	N=10000,p=100		N=200,p=10000	
Correlación	0	0.5	0	0.5
Descenso por coordenadas	0.110(0.001)	0.127(0.002)	0.298(0.003)	0.513(0.014)
Nesterov	0.251(0.007)	0.604(0.011)	1.555(0.049)	2.914(0.119)

Cuadro 6.1: LASSO para la regresión lineal: promedio de tiempos de una CPU en diez realizaciones, para el descenso por coordenadas y Nesterov. En cada caso, el tiempo mostrado es el tiempo total en una ruta de 20 λ valores.

	N=10000,p=100		N=200,p=10000	
Correlación	0	0.5	0	0.5
Descenso por coordenadas	0.309(0.086)	0.306(0.086)	0.646(0.006)	0.882(0.026)
Nesterov	1.482(0.020)	2.867(0.045)	2.910(0.106)	8.292(0.480)

Cuadro 6.2: LASSO para la regresión lineal: promedio de tiempos de una CPU en diez realizaciones, para el descenso por coordenadas y Nesterov. En cada caso, el tiempo mostrado es el tiempo total en una ruta de 20 λ valores.

Las predicciones para la tabla siguiente fueron generadas como antes, pero ahora hay 15 coeficientes β_j distintos de cero con signos alternativos, y $|\beta| = 15 - j + 1$. Luego, definiendo $p_i = 1/(1 + \exp(-\sum \beta_j x_{ij}))$ generamos $0/1Y_i$ con $P(y_i = 1) = p_i$. Se puede ver que el descenso por coordenadas es de 5 a 10 veces más rápido que el de Nesterov, con una mayor aceleración en el caso $p > N$.

6.3. Regresión LASSO Bayesiana

El **paradigma Beyessiano** trata los parámetros como cantidades aleatorias junto con una distribución previa que caracteriza nuestra creencia en lo que sus valores podrían ser. Adoptaremos el enfoque de Park y Casella [**17**], que nos sugiere un modelo de la forma

$$\mathbf{y}|\beta, \lambda, \sigma \sim N(\mathbf{X}\beta, \sigma^2 \mathbf{I}_{N\times N}) \tag{6.21}$$

$$\beta|\lambda, \sigma \sim \prod_{j=1}^{p} \frac{\lambda}{2\sigma} e^{-\frac{\lambda}{\sigma}|\beta_j|}, \tag{6.22}$$

usando que son independientes e idénticamente distribuidas. Bajo este modelo, es fácil demostrar que la densidad posterior logarítmica negativa para $\beta|\mathbf{y}, \lambda, \sigma$ viene dada por:

$$\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \frac{\lambda}{\sigma}\|\beta\|_1, \tag{6.23}$$

donde hemos dejado caer una constante positiva independiente de β. Consecuentemente, para cualquier valor fijado de λ y σ, el modo posterior coincide con el estimador de LASSO. Hemos asumido que no hay constante en el modelo, y que las columnas de $\mathbf{X}$ están centradas en la media, al igual que $\mathbf{y}$.

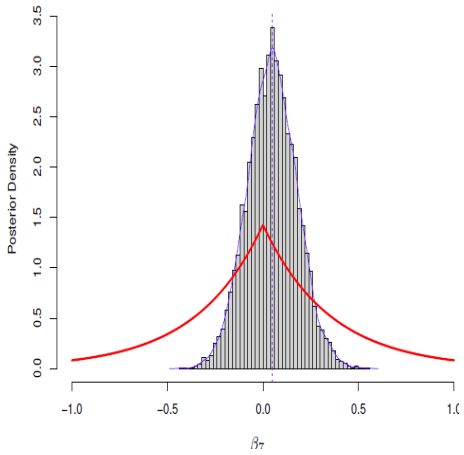

Figura 6.3: Distribución previa y posterior para la séptima variable en el ejemplo de diabetes, con λ mantenido fijo. La distribución previa en la figura es una distribución doble exponencial (Laplace) con densidad proporcional a $\exp(-6{,}5 \cdot 10^{-3}|\beta_7|)$.

La distribución posterior nos proporciona más que estimaciones puntuales: proporciona una distribución conjunta completa.

La curva roja en la Figura 6.3 *"Laplace prior"* usada en el LASSO Bayesiano, aplicado a la variable β_7 en los datos de diabetes. Estos datos consisten en la observaciones de 442 pacientes, siendo la respuesta de interés una medida cuantitativa de la progresión de la enfermedad un año después de ser detectada. Hay diez variables de referencia (edad, sexo, índice de masa corporal, presión media sanguínea y seis mediciones de suero sanguíneo) más términos cuadráticos, lo que da un total de 64 características. La distribución previa tiene un pico en 0, lo que refleja nuestra creencia de que algunos parámetros son cero. Dada una distribución de probabilidad para los datos observados dados los parámetros, actualizamos nuestra distribución previa condicionando los datos observados, produciendo la distribución posterior de los parámetros.

La distribución previa tiene un parámetro de varianza que caracteriza la fuerza de nuestra creencia en que cero es un valor especial. El modo posterior está ligeramente alejado de cero, aunque un intervalo creíble posterior del 95 % cubre cómodamente a cero. Cálculos exactos son casi siempre intratables, a excepción de los modelos más simples. Afortunadamente, los cálculos modernos nos permiten usar *Marko chain Monte Carlo (MCMC)* (cadena de Markov Monte Carlo) para muestrear de mane-

ra eficiente las realizaciones de las distribuciones posteriores de los parámetros de interés.

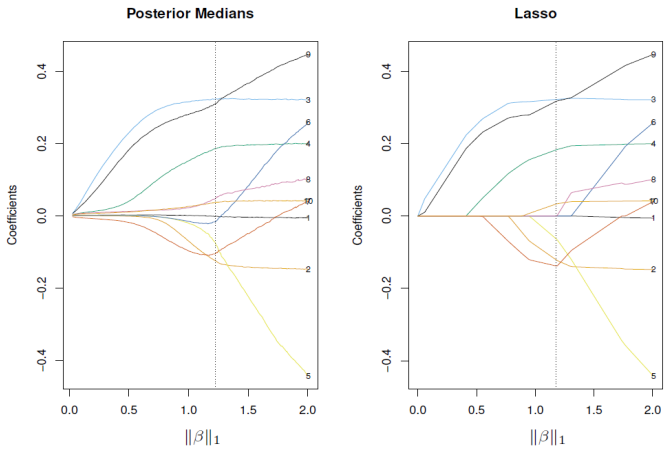

Figura 6.4: LASSO Bayessiano en los datos de diabetes. La figura de la izquierda muestra las medianas posteriores ejecutadas por MCMC (condicionadas en λ). Por otro lado, la figura de la derecha muestra el perfil de LASSO. En la de la izquierda, la línea vertical está en mediana posterior de $\|\beta\|_1$, mientras que en la derecha, la línea vertical se encontró mediante una validación cruzada de N veces.

La figura 6.4 (panel izquierdo) muestra un resumen de muestras de MCMC de la distribución posterior de $\beta|\lambda$; la mediana de 10000 muestras posteriores se muestra en cada 100 valores de λ. Aquí, σ^2 se permite variar (con $\pi(\sigma^2) \sim \frac{1}{\sigma^2}$). Esto, y el hecho de que hemos mostrada medianas, explica las leves discrepancias con el gráfico de la derecha (LASSO), que muestra el modo posterior para valores fijados de $\sigma\lambda$. Un modelo Bayesiano completo también especificará a una distribución previa para λ; ene este caso, una distribución Gamma difusa es conjugada, y por lo tanto, conveniente para el muestreo de MCMC.Aquí es donde el enfoque Bayesiano puede hacer valer la pena el esfuerzo extra considerable y el acto de fe. La distribución posterior completa incluye λ tanto como β, por lo que esa selección del modelo se realiza de forma automática. Además, los intervalos creíbles posteriores para β tienen en cuenta la variabilidad posterior en λ.

La Figura 6.5 (que viene a continuación) muestra un resumen de 10000 MCMC muestras de la distribución posterior de los datos de diabetes. Mientras el modo posterior tiene nueve coeficientes distintos de cero, la distribución posterior sugiere que son entre 5 y 8 de éstos están bien separados del cero.

Especificar el modelo Bayesiano es un desafío técnico, y hay numerosas elecciones que tomar a lo largo del camino: éstos incluyen a priori para λ y σ^2, que a su vez tienen hiperparámetros que deben establecerse.

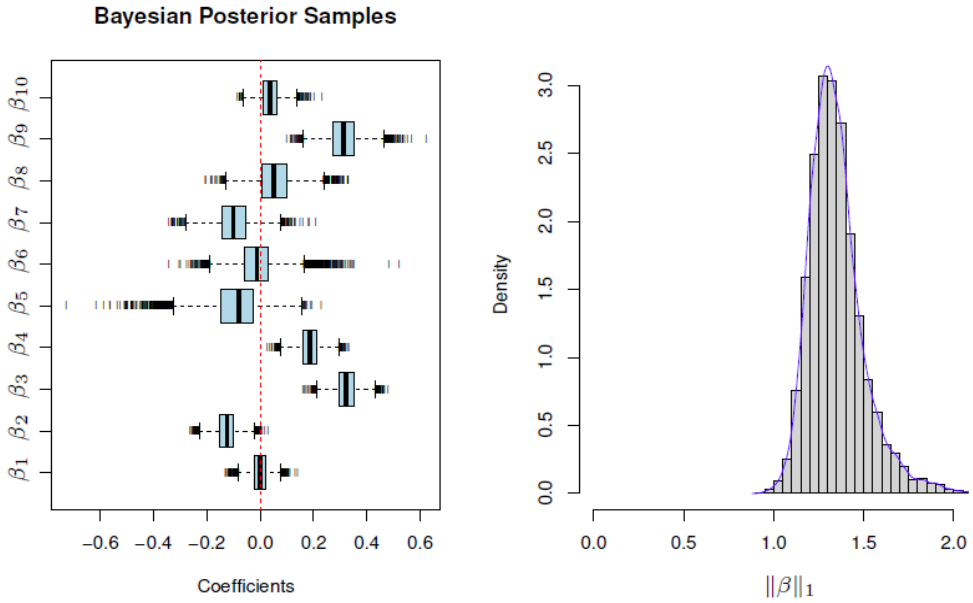

Figura 6.5: Distribuciones posteriores para β_j y $\|\beta\|_1$ para los datos de diabetes. Resumen de 10000 muestras de MCMC, con las 1000 primeras muestras "quemadas" descartadas.

6.4. Bootstrap

Bootstrap es una herramienta no paramétrica popular para evaluar las propiedades estadísticas de estimadores complejos. Para motivar su uso, supongamos que hemos obtenido un estimador $\widehat{\beta}(\widehat{\lambda}_{CV})$ para un problema LASSO acorde al siguiente procedimiento:

- Ajustar un camino LASSO para $(\mathbf{X}, \mathbf{y})$ sobre una densa cuadrícula de valores $\Lambda = \{\lambda_l\}_{l=1}^{L}$.
- Dividir las muestras de aprendizajes en 10 grupos de manera aleatoria.
- Con el grupo k que hemos dejado fuera, ajustar un camino LASSO con los 9/10 grupos restantes, usando la misma cuadrícula Λ.
- Para cada $\lambda \in \Lambda$ calcular el error de predicción de media cuadrática para el grupo que hemos dejado fuera.
- Promediar estos errores para obtener una curva del error de predicción sobre la cuadrícula Λ.
- Encontrar el valor $\widehat{\beta}(\widehat{\lambda}_{CV})$ que minimiza la curva, y luego devolver nuestro vector de coeficientes de nuestro ajuste en el primer paso en el valor de λ.

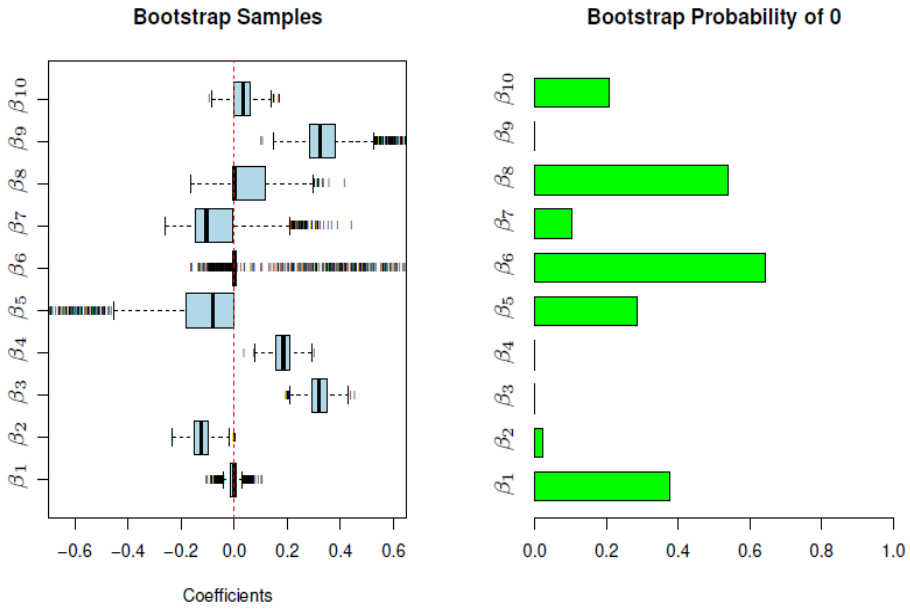

Figura 6.6: [Izquierda] Boxplots de 1000 bootstrap realizaciones de $\widehat{\beta}^*(\widehat{\lambda}_{CV})$ obtenido por el bootstrap no paramétrico, que corresponde a la f.d.D. empírica $\widehat{F}_N$. Comparando con la correspondiente distribución Bayesiana posterior en la figura anterior, vemos una estrecha correspondencia en este caso. [Derecha] Proporción de veces que cada coeficiente es cero en la distribución bootstrap.

¿Cómo evaluamos la distribución muestral de $\widehat{\beta}^*(\widehat{\lambda}_{CV})$? Pues bien, estamos interesados en la distribución del estimador aleatorio $\widehat{\beta}^*(\widehat{\lambda}_{CV})$ como una función de N muestras independientes e idénticamente distribuidas $\{(x_i, y_i)\}_{i=1}^N$. El bootstrap no paramétrico es un método para aproximar estas distribuciones muestrales: para ello, se aproxima a la función de distribución acumulada F del par aleatorio (X, Y) por función de distribución acumulada $\widehat{F}_N$ definida por N muestras. Luego extraemos N muestras de $\widehat{F}_N$, lo que equivale a extraer N muestras con remplazo del conjunto de datos dado.

Figura 6.6 [izquierda] muestra los boxplots de 1000 realizaciones bootstrap de $\widehat{\beta}^*(\widehat{\lambda}_{CV})$ obtenidos de esta manera, repitiendo los pasos 1-6 en cada muestra bootstrap. Hay una correspondencia razonable entre esta figura, los resultados Bayesianos correspondientes en la figura 6.5 . La gráfica de la derecha muestra una proporción del número de veces que cada variable fue exactamente cero en la distribución bootstrap. Ninguno de las realizaciones Bayesianas posteriores son exactamente cero, aunque con frecuencia algunas se acercan a cero.

La figura 6.7 muestra las curvas de validación cruzada "bootstrapped", y sus mínimos. Los mínimos bootstrapped tienen una amplia propagación ya que la curva CV original es plana en una amplia región. Curiosamente, las bandas de error estándar de bootstrap guardan una estrecha relación con las calculadas a partir del ajuste de CV original en el gráfico de la izquierda. La figura que veremos a continuación, muestra un diagrama de los coeficientes del bootstrapped. Para tales gráficas, podemos ver cómo las variables correlacionadas pueden intercambiarse entre sí, tanto en valor como en su tendencia a ser cero.

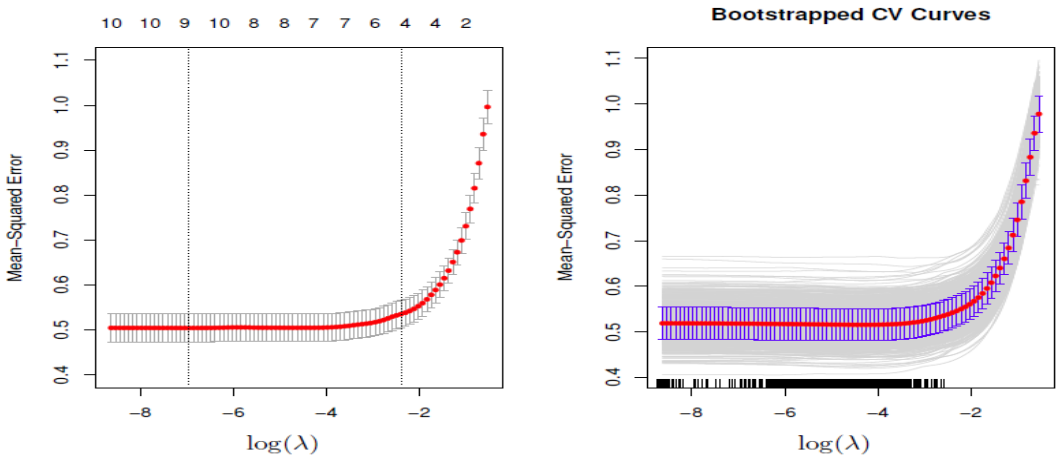

Figura 6.7: [Izquierda] La curva de validación cruzada para LASSO en los datos de diabetes, con un error estándar calculado a partir de 10 realizaciones. La línea vertical en la izquierda corresponde al valor minimizante de λ. La línea de la derecha corresponde a una regla de error estándar; el mayor valor de λ para el cual, el error CV está dentro de un error estándar del valor de minimización. [Derecha] 1000 curvas CV bootstrap, con la media en rojo y bandas de error estándar en azul. El "rug-plot" (es un gráfico de datos para una sola variable cuantitativa, que se muestra como marcas a lo largo de un eje; se utiliza para visualizar la distribución de los datos) de la base muestra la localización de los mínimos.

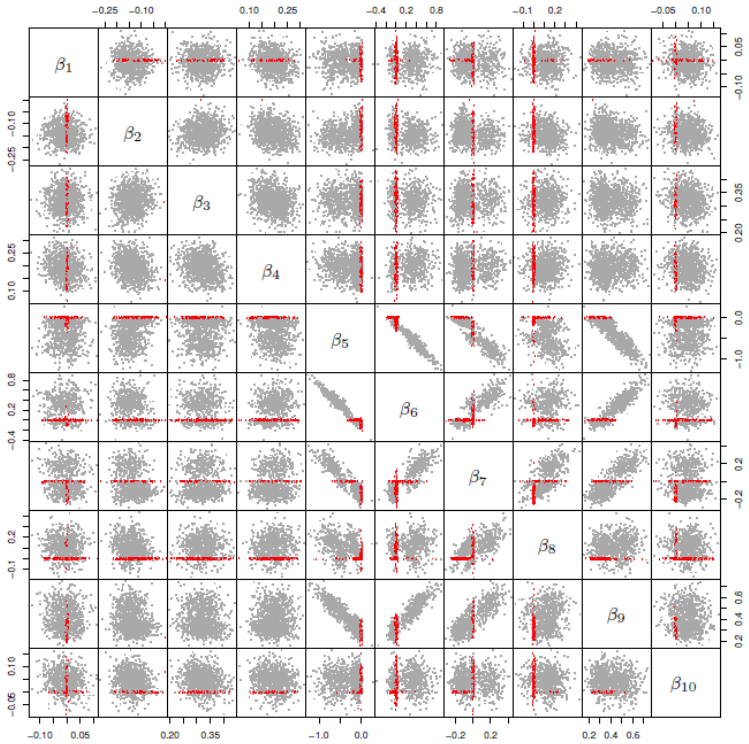

Figura 6.8: Las gráficas por parejas de los coeficientes bootstrapped $\widehat{\beta}^*(\widehat{\lambda}_{CV})$. Los puntos rojos de la figura 6.8 corresponden a los valores que son cero en al menos una coordenada para esa gráfica. La muestras x_5 y x_6 tienen una gran correlación; se ve además la correspondiente correlación negativa en sus coeficientes, jugando cero un papel importante.

En la tabla que veremos a continuación, mostramos los tiempos comparativos en segundos para problemas con N=400 y diferentes número de predicciones. Generamos 1000 muestras bootstrap; para LASSO Bayesiano generamos 2000 muestras posteriores, con la idea de descartar la 1000 primeras muestras. Mientras que tales comparaciones dependen de detalles de implementación, el crecimiento relativo de p es informativo. Quizá LASSO Bayesiano es más rápido para problemas pequeños, pero su complejidad parece ser de la escala del orden de $O(p^2)$. Por el contrario, la escala de bootstrap parece estar cerca del orden de $O(p)$, porque aprovecha la escasez y la convexidad de LASSO.

El procedimiento que usamos usó el *bootstrap no paramétrico*, en el cual, estimamos la población desconocida F por la función de distribución empírica $\widehat{F}_N$, la estimación de máxima verosimilitud no paramétrica de F. El muestreo de $\widehat{F}_N$ corresponde al muestreo con el reemplazo de los datos. En contraste, las el *bootstrap paramétrico* toma muestras de una estimación paramétrica de F, o su correspondiente función de densidad f. En este ejemplo, arreglaríamos X y obtendríamos estimaciones $\widehat{\beta}$ y $\widehat{\sigma}^2$, ya sea del ajuste de mínimos cuadrados completo, o de LASSO ajustado con el parámetro λ.

p	LASSO Bayesiano	LASSO/Bootstrap
10	3.3 seg	163.8 seg
50	184,8 seg	374,6 seg
100	28,6 min	14,7 min
200	4,5 h	18,1 min

Cuadro 6.3: Tiempo para LASSO Bayesiano y bootstrapped LASSO, para 4 problemas de diferentes tamaños con N=400.

Usando los estimadores del ajuste por mínimos cuadrados $\widehat{\beta}$ y $\widehat{\sigma}^2$, los resultados para el bootstrap paramétrico de nuestro ejemplo lo mostramos en la figura que viene a continuación. Son similares tanto a los resultados de bootstrap no paramétricos como a los del LASSO Bayesiano. En general, cabría esperar que el bootstrap paramétrico probablemente produciría resultados inclusos más cercanos al LASSO Bayesiano comparado con el bootstrap no paramétrico, a que el bootstrap paramétrico y el LASSO Bayessiano usan la forma paramétrica asumida (6.21). Notemos además que, el uso de los estimadores del ajuste por mínimos cuadrados $\widehat{\beta}$ y $\widehat{\sigma}^2$ podrían no funcionar cuando $p \gg N$, y necesitaríamos unos datos diferentes para cada valor de λ. Esto ralentizaría los cálculos considerablemente.

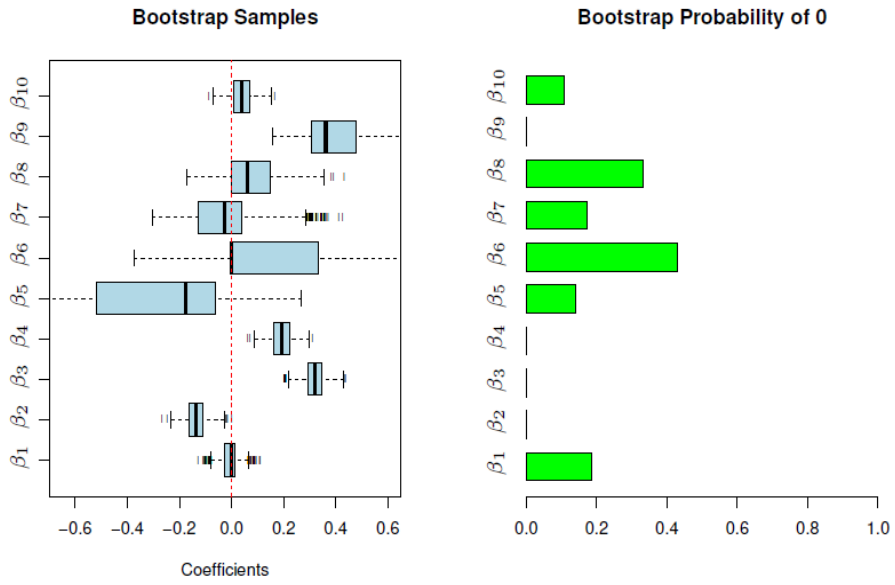

Figura 6.9: [Izquierda] Boxplot de 1000 realizaciones bootstrap paramétricas de $\widehat{\beta}^*(\widehat{\lambda}_{CV})$. Comparando con la distribución posterior bayessiana correspondiente en la figura 6.5, nuevamente vemos una estrecha correspondencia. [Derecha] Proporción de tiempo en que cada coeficiente es cero en la distribución bootstrap.

Capítulo 7

Conclusión

Bien es cierto que más de uno podría argumentar que la optimización convexa no debería resultar tan interesante para el aprendizaje automático, ya que a menudo encontramos superficies que se encuentran lejos de ser convexas. Sin embargo, la optimización convexa presenta multitud de ventajas como la rapidez y el ser más simple, además de menos costosa computacionalmente; por ejemplo, el método de descenso por gradiente (y algunos algoritmos derivados de él) se usan con frecuencia en el aprendizaje automático, y principalmente paras redes neuronales, porque se ha demostrado que funcionan, son escalables y podemos implementarlos en multitud de programas.

Este trabajo se plantea con el objetivo de abarcar el *boom* que lleva teniendo el aprendizaje automático desde principios de siglo con el tema principal de los problemas de optimización convexa, pues bien es cierto que a día de hoy cualquier problema de optimización resoluble es de este tipo, y por tanto, se hace necesario tener unos conceptos en torno a convexidad y las propiedades que nos aporta en la resolución de dichos problemas. Además, el éxito a la hora de resolver problemas de aprendizaje automático va a estar principalmente determinado por la capacidad de obtener una solución en el menor tiempo posible, que entendemos como la capacidad de dar una respuesta (una solución) frente a un problema práctico, con unos datos dados, por lo que no sólo basta desarrollar algoritmos teóricamente aptos pero luego difíciles o imposibles de implementar, y es esto por tanto, lo que nos conduce a los problemas de optimización convexa.

En el capítulo 4 hemos introducido el método de descenso por gradiente, junto a otros métodos alternativos, y esto es porque dichos métodos presentan una gran facilidad a la hora de ser implementados y cuya convergencia es independiente de la dimensión en las que nos encontremos. Además, obtenemos una serie de resultados sobre las cotas que nos van a garantizar, bajos ciertas hipótesis sobre nuestra función objetivo f, una velocidad elevada para cuando queramos obtener una solución lo suficientemente buena, es decir, próxima al mínimo. Bien es cierto, que en algunos casos, las hipótesis previas serán un poco idealizadas, por lo que esto nos llevaría a tratar con funciones quizá no convexas y por tanto, a aproximar nuestro problema no convexo a uno convexo; pero no va a ser el caso, nosotros vamos a concretizar un poco más nuestro problema, tomando como ejemplo de estudio el problema LASSO.

En el capítulo 6 hemos introducido el problema LASSO como una penalización sobre el problema de mínimos cuadrados de manera que, posteriormente hemos hablado del método de descenso por coordenadas como una forma de resolver dicho problema pero presenta el siguiente inconviente: es lento debido a que tiene que actualizar coordenada a coordenada el estimador y junto a que, la función objetivo vista carece de una propiedad que nos vendría genial como es β-suave, se nos hace difícil, a priori, una mejor manera de resolverlo. Pero es aquí cuando nos percatamos de que dicha función la podemos dividir en dos funciones, la primera de ellas convexa y suave, y la segunda convexa.

Por otro lado, aparece el método de Nesterov, que a simple vista se nos presenta con un enfoque simple y computacionalmente bueno, por lo que nos planteamos una comparación entre éste y el descenso por coordenadas. Para ello, hicimos un pequeño estudio de simulación en el que acabamos concluyendo que el descenso por coordenadas es de 5 a 10 veces más rápido que el de Nesterov y además, en el caso de que p sea mayor que N la aceleración de éste es mayor. También vemos el modelo Bayessiano, pero especificarlo como tal es un desafío técnico con numerosas elecciones que tomar a lo largo del camino, ya que éste trata los parámetros como cantidades aleatorias junto con una distribución previa. Por tanto es aquí donde aparece el bootstrap.

El bootstrap fue propuesto por el estadístico Bradley Efron y nos permite evaluar propiedades estadísticas sobre estimadores complejos, pues como hemos dicho, no todos los problemas de la vida real son problemas de optimización convexa, por lo que, los estimadores que usamos, en la regresión por ejemplo, no van a ser tampoco tan ideales como quisiéramos. Aunque pueda parecer un procedimiento muy complejo (descrito en la última sección del capítulo 6), dicho proceso en el que se basa es simplemente la creación de un gran número de muestras reposicionando los datos tomando como referencia una muestra poblacional inicial, lo que va a resultar muy útil en los casos en que nos encontremos con muestras pequeñas o que la distribución sea muy sesgada.

Por último y para concluir, espero que el trabajo haya resultado útil con el fin de consolidar la idea de que, la optimización convexa, resulta fundamental en el campo del aprendizaje automático, pues nos permite alcanzar algoritmos rápidos y escalables, con todo un frente abierto para seguir investigando dada su importancia en la inteligencia artificial.

Bibliografía

[1] BOYD, A. y VANDERBERGHUE, L., *Convex Optimization.* Cambrigde University Press, 2004.

[2] TIKHONOV, A. N. y ARSENIN, V. Y., *Solution of Ill-Posed Problems.* V. H. Winston, Whashington, DC, 1977.

[3] BECK, A. y TEOBULLE, M., *A Fast Iterative Shrinkage-Theresholding Algorithm for Linear Inverse Problems.* SIAM Journal on Imaging Sciences, 2(1): p. 183-202, 2009.

[4] BUBECK, S., *Convex Optimization: Alogrithms and Complexity.* Theory Group, Microsoft Research, 2015.

[5] BEN-TAL, A. y NEMIROVSKI, A., *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications.* MPS/SIAM Ser.Optim, SIAM, Philadelphia, 2001.

[6] NICKEL, S., *Convex Analysis.* Department of Mathematics. University of Kaiserslautern, 1998.

[7] HASTIE, T., TIBSHIRANI, R. y WAINWRIGHY, M., *Statistical Learning with Sparsity: The LASSO and Generalizations.* Taylor and Francis Group. CRC Press, 2015.

[8] BRUCK, R. J., *On the weak convergence of an ergodic iteration for the solution of varational inequalities for monotone operators in Hilberts space.* J. Math. anal., 61, p. 159-164, 1977.

[9] PASSTY, G. B., *Ergodic convergence to a zero of the sum of monotone operators in Hilbert space.* J. Math. Ana. Appl., 72, p. 383-390, 1979.

[10] ELAD, M., MATALON, B. y ZIBULEVSKY, M., *Subspace optimization methods for linear least squares with non-quadratic regularization.* Appl. Comput. Harmon. Anal., 23, p. 346-367, 2007.

[11] NEMIROVSKY, A. S. y YUDIN, D. B., *Problem Complexity and Method Efficiency in Optimization,Wiley-Interscience Series in Discrete Mathematics.* John Wiley and Sons, New York, 1983.

[12] NESTEROV, Y. E., *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$. Dokl. Akad. Nauk SSSR, 269, p. 543-547 (Ruso), 1983.

[13] SU, W., BOYD, S. y CANDLÈS, E., *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*. In Advances in Neural Information Processing Systems (NIPS), 2014.

[14] ALLEN-ZHU, Z. y ORECCHIA, L., *Linear coupling: An ultimate unification of gradient and mirror descent*. Arxiv preprint arXiv:1407.1537, 2015.

[15] TSENG, P., *Coordinate ascent for maximizing nondifferentiable concave function*. Technical Report LIDS-P: 1840, Massachusets Institute of Technology. Laboratory for Information and Decsion Systems, 1988.

[16] TSENG, P., *Convergence of block coordinate descent method for nondifferentiable maximization*. Journal of Optimization Theory and Applications **109**(3), p. 474-494, 2001.

[17] PARK, T. y CASELLA, G., *The Bayessian Lasso*. Journal of the American Statistical Association **103**(482), p. 681-686, 2008.