Sección de Matemáticas
Universidad de La Laguna

Julia Alicia Palenzuela Rodríguez

Problemas de Programación Convexa

Convex Programming Problems

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Mayo

Dirigido por
Carlos González Martín

Carlos González Martín
Matemáticas, Estadística e
Investigación Operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A quien me acompañó en el camino sin dudarlo.

Julia Alicia Palenzuela Rodríguez
La Laguna, 20 de mayo de 2023

Resumen · Abstract

Resumen

La optimización convexa es un área importante de las matemáticas que trata problemas con múltiples aplicaciones en muchos campos de la ciencia y la ingeniería. El objetivo de esta memoria es proporcionar un estudio de problemas de optimización convexa, incluyendo aspectos teóricos y metodológicos sobre los que se basan distintas aplicaciones. Para lograr este objetivo, se comienza con la definición de conceptos básicos de la teoría de conjuntos y funciones convexas. A continuación, se profundiza en los problemas de optimización convexa y sus condiciones de optimalidad. Se explican diferentes métodos de resolución, con ejemplos ilustrativos. Finalmente, se ejemplifica la aplicación de los conceptos y métodos estudiados en diferentes situaciones, con el objetivo de proporcionar al lector una comprensión completa y útil del tema.

Palabras clave: *Optimización – Convexidad – Diferenciable – KKT*

Abstract

Convex optimization is an important area of mathematics, dealing with problems with multiple applications in many fields of science and engineering. The aim of this dissertation is to provide a study of convex optimization problems, including theoretical and methodological aspects on which different applications are based. To achieve this objective, the dissertation begins defining basic concepts within the theory of convex sets and functions. Then, it delves into convex optimization problems and their optimality conditions. Various resolution methods are explained with illustrative examples. Finally, the application of the concepts and methods studied in different situations is exemplified, with the aim of providing the reader with a comprehensive and useful understanding of the topic.

Keywords: *Optimization – Convexity – Differentiable – KKT*

Contenido

Introducción

En el estudio de la Optimización, una vez explorado el caso lineal, los problemas de programación convexa pueden ser considerados el siguiente punto de interés antes de profundizar en el amplio campo de la programación no lineal al que pertenecen. En los problemas de optimización convexa se minimiza (maximiza) una función convexa (cóncava) en un conjunto convexo. Esto incluye muchos problemas de optimización no lineal, como la minimización de funciones cuadráticas y la optimización de sistemas dinámicos. Además, muchas técnicas de optimización convexa son más eficientes para resolver problemas de optimización no lineales. Por lo tanto, elegir la optimización convexa puede ser beneficioso en situaciones donde se requiere una formulación más general o cuando se necesitan soluciones más eficientes para problemas no lineales.

En los últimos años, ha habido un gran avance en el estudio de la optimización convexa y la convexidad, tanto en términos de teoría como de aplicaciones prácticas. Esto se debe al rápido desarrollo del área de ciencia de datos y el aprendizaje automático, que hace un gran uso de estas técnicas para la resolución de problemas de clasificación y regresión.

En esta memoria, relacionaremos los conceptos básicos necesarios de la programación convexa.

En el capítulo 1, introducimos el tema relacionando conceptos y propiedades básicas de los conjuntos convexos y de las funciones convexas como paso previo imprescindible para avanzar y profundizar en los problemas que queremos estudiar en esta memoria. Se enfoca en las definiciones de conjuntos, funciones convexas y funciones convexas generalizadas.

El capítulo 2 vemos el problema general de optimización convexa. También estudiamos las condiciones de optimalidad, tanto para funciones convexas como para funciones convexas generalizadas. Por último, generalizamos a problemas de programación no lineal y se introduce la definición de punto KKT.

En el capítulo 3 presentamos algunos métodos que se pueden usar para resolver los problemas introducidos anteriormente, incluyendo el Método de Gradiente Reducido, el Método Simplex Convexo de Zangwill y el Método de

Dirección Alterna de Mutiplicadores. Se describen en detalle los algoritmos y se acompañan con ejemplos.

Finalmente, en el capítulo 4, se presentan algunos ejemplos de problemas de optimización convexa en diversas aplicaciones, incluyendo problemas de mínimos cuadrados, problemas cuadráticos, problemas semidefinidos, programación geométrica y problemas de control. Se describen los modelos matemáticos para cada problema y se presentan los métodos de resolución correspondientes, con ayuda del leguaje de programación Python.

1

Conjuntos convexos y funciones convexas

En este tema se relacionan conceptos y propiedades básicas de los conjuntos convexos y de las funciones convexas como paso previo imprescindible para el estudio y resolución de problemas de optimización convexa. Las definiciones siguientes las podemos encontrar en [1] y [3].

1.1. Conjuntos convexos

Un subconjunto $S \subseteq R^n$ es convexo si, y solo si, contiene los segmentos de recta que conectan cualquier par de puntos, es decir:

$$\forall x_1, x_2 \in S, \forall \theta \in [0,1], x_1 + \theta(x_2 - x_1) \in S$$

Vemos algunos ejemplos de conjuntos convexos, encontrados en la sección 2 de [3]

1.1.1. Ejemplos

- $S = \{(x_1, x_2, x_3) : x_1 + 2x_2 - x_3 = 4\} \subset R^3$

Este conjunto es un ***plano*** en R^3. En general, $S = \{x : p^t x = \theta\}$ se le llama hiperplano en R^n, donde p es un vector no nulo en R^n, llamado *gradiente* del hiperplano, y θ es un escalar. Un hiperplano divide el espacio R^n en dos semiespacios.

- ***Poliedros convexos***

Un conjunto $P \subseteq R^n$ se llama ***poliedro*** si es la intersección de un número finito de semiespacios cerrados; es decir, $P = \{x : p_i^t x \leq \alpha_i$, para $i = 1, \ldots, m\}$ donde p, es un vector distinto de cero y α es un escalar para $i = 1, \ldots, m$.

Por ejemplo, un poliedro es:

$$S = \{(x, y) : -x + y \leq 2, y \leq 4, x \geq 0, y \geq 0\}$$

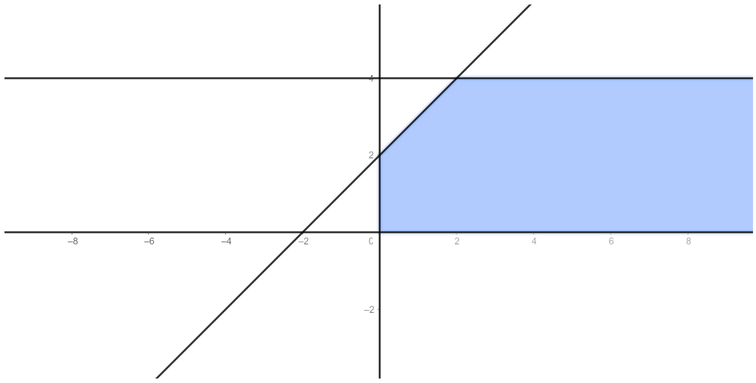


Figura 1.1: Gráfica de un poliedro

Gráficamente,

Los poliedros son, obviamente, conjuntos convexos. Lo contrario no es cierto.

- ***Entornos euclídeos y elipsoides***

Un entorno euclídeo en R^n tiene la forma

$$N(x_c, r) = \{x/\|x - x_c\|_2 \leq r\} = \{x/(x - x_c)^T(x - x_c) \leq r^2\}$$

donde $r > 0$ y $\|.\|_2$ denota la norma euclidiana. El vector x_c es el centro, y el escalar r es su radio. El entorno $N(x_c, r)$ consta de todos los puntos que distan una distancia r del centro.

Un entorno euclídeo es un conjunto convexo:

Si $\|x_1 - x_c\|_2 \leq r$, $\|x_2 - x_c\|_2 \leq r$, $\forall\theta \in [0, 1]$, entonces:

$$\begin{aligned}
&\|\theta x_1 + (1-\theta)x_2 - x_c\|_2 = \\
&= |\theta(x_1 - x_c) + (1-\theta)(x_2 - x_c)\|_2 \\
&\leq \theta\|x_1 - x_c\|_2 + (1-\theta)\|x_2 - x_c\|_2 \\
&\leq r
\end{aligned}$$

Otra familia de conjuntos convexos son los elipsoides, que tienen la forma

$$\epsilon = \{x : (x - x_c)^T L^{-1}(x - x_c) \leq 1\}$$

El vector $x_c \in R^n$ es el centro de la elipsoide. La matriz $L \in S^n_+$ determina la extensión del elipsoide; la longitud de los semiejes de ϵ se calcula a partir de los autovalores de P como $\sqrt{\lambda_i}$. Un entorno euclídeo es un elipsoide con $L = r^2 I$.

Definición

Un conjunto S $\subseteq V$ es afín sí

$$\forall x_1, x_2 \in S, \forall\theta \in R, \theta x_1 + (1-\theta)x_2 \in A$$

1.1.2. Propiedades

- Un conjunto afín es convexo. Lo contrario no es cierto.
- Cualquier semiespacio es un conjunto convexo.
  Si S_1 y S_2 son convexos:
- $S_1 \cap S_2$ es convexo.
- $S_1 \oplus S_2 = \{x_1 + x_2 : x_1 \in S_1, x_2 \in S_2\}$ es convexo.
- $S_1 \ominus S_2 = \{x_1 - x_2 : x_1 \in S_1, x_2 \in S_2\}$ es convexo.

1.2. Algunos conceptos y resultados básicos sobre conjuntos convexos. Teoremas de la alternativa

1.2.1. Combinación lineal convexa

Dados k puntos de R^n $x_1, \ldots, x_k$, y escalares $\theta_1, \ldots, \theta_k \in [0,1]$, $\sum_{i=1}^{k} \theta_i = 1$, diremos que $\sum_{i=1}^{k} \theta_i x_i$ es una combinación lineal convexa de $x_1, \ldots, x_k$.

Proposición

Un conjunto convexo contiene las combinaciones lineales convexas formadas con puntos de ese conjunto.

Nótese que la proposición anterior no puede extenderse a cualquier subconjunto de R^n: Para los subconjuntos que no son convexos habrán combinaciones lineales convexas de sus puntos que no pertenecen a dichos subconjuntos. Precisamente, para determinar el menor de los conjuntos convexos que contiene a un subconjunto dado, se da la siguiente definición.

1.2.2. Envoltura convexa

Dado un conjunto $S \in R^n$, la envoltura convexa de S, $H(S)$ es el conjunto de todas las combinaciones lineales convexas formados con puntos de S.

1.2.3. Teorema de Carathéodory

Por definición, un punto en la envoltura convexa de un conjunto se puede representar como una combinación convexa de un número finito de puntos en el conjunto. El siguiente teorema muestra que cualquier punto x en la envoltura convexa de un conjunto S se puede representar como una combinación convexa de, como máximo, $n+1$ puntos en S.

Teorema

Sea S un conjunto en R^n. Cada punto en la envoltura convexa de S es una combinación convexa de un subconjunto afínmente independiente de S. En particular, es una combinación convexa de como máximo $n+1$ puntos de S. En otras palabras, $x \in S$ puede representarse como:

$$x = \sum_{j=1}^{n+1} \theta_j x_j,\ \sum_{j=1}^{n+1} \theta_j = 1$$
$$\theta_j \geq 0 \text{ para } j = 1, \ldots, n+1$$
$$x_j \in S \text{ para } j = 1, \ldots, n+1$$

1.2.4. Interior y clausura de un conjunto

Sea S un conjunto arbitrario en R^n. Se dice que un punto x está en la *clausura de* S, denotada por $Cl(S)$ si $S \cap N_\varepsilon(x) \neq \emptyset$, $\forall \varepsilon > 0$. Si $S = Cl(S)$, S es un conjunto cerrado.

Se dice que un punto x está en el *interior de* S, denotado por $Int(S)$, si $N_\varepsilon(x) \subset S$ para algún $\varepsilon > 0$. Un conjunto *sólido* $S \subseteq R^n$ es aquel que su interior es no vacío. Si $S = int(S)$, entonces S es un conjunto abierto.

Finalmente, x se dice que está en la *frontera* de S, denotado como $Fr(S)$, si $N_\varepsilon(x)$, $\forall \varepsilon > 0$ contiene al menos un punto en S y un punto fuera de S.

Un conjunto S está *acotado* si puede estar contenido en un entorno de un radio suficientemente grande. Un conjunto *compacto* es aquel que es a la vez cerrado y acotado. Debemos tener en cuenta que el complementario de un conjunto abierto es un conjunto cerrado (y viceversa), y que los puntos frontera de cualquier conjunto y su complementario son los mismos.

Propiedades

- El complementario de un conjunto abierto, es un conjunto cerrado (y viceversa).
- Sea $S \in R^n$ un conjunto convexo con $Int(S)$ no vacío. Sea $X_1 \in Cl(S)$ y $x_2 \in Int(S)$. Entonces $\theta x_1 + (1-\theta)x_2 \in Int(S)$, $\forall \theta \in (0, 1)$.
- Si S es un conjunto convexo, entonces $Int(S)$ es convexo.
- Sea $S \in R^n$ un conjunto convexo con $Int(S)$ no vacío. Entonces, $Cl(S)$ es convexo.
- Sea $S \in R^n$ un conjunto convexo con $Int(S)$ no vacío. Entonces, $Cl((Int(S))$= Cl(S)

1.2.5. Soporte y separación

Las nociones de soporte de hiperplanos y separación de conjuntos convexos disjuntos son muy importantes en la optimización. Casi todas las condiciones de optimalidad y las relaciones de dualidad utilizan algún tipo de separación o soporte de conjuntos convexos.

Distancia mínima de un punto a un conjunto convexo

Dado un conjunto convexo cerrado $S \subseteq R^n$ no vacío, y $y \notin S$. Entonces, existe un único $\overline{x} \in S$ con mínima distancia respecto de y.

Hiperplanos y separación de dos conjuntos

Sean $S_1, S_2 \subseteq R^n$ conjuntos no vacíos. Un hiperplano $H = \{x : p^t x = \alpha\}$ se dice que separa a S_1 y S_2 si se cumple $p^t x \geq \alpha$ para cada $x \in S_1$ y $p^t x \leq \alpha$ para cada $x \in S_2$.

Además, si se cumple que $S_1 \cup S_2 \not\subset H$, se dice que H separa *correctamente* a S_1 y S_2. Se dice que H separa *estrictamente* a S_1 y S_2 si $p^t x > \alpha$ para cada $x \in S_1$ y $p^t x < \alpha$ para cada $x \in S_2$. Por último, se dice que H separa *fuertemente* a S_1 y S_2 si $p^t x \geq \alpha + \varepsilon$ para cada $x \in S_1$ y $p^t x \leq \alpha$ para cada $x \in S_2$, con $\varepsilon > 0$.

Separación de un conjunto convexo y un punto

Presentamos ahora el primer y más importante teorema de separación.

Teorema

Sea $S \in R^n$ un conjunto cerrado no vacío, y $y \notin S$. Entonces, $\exists p$ vector no nulo y α escalar tal que $p^t y > \alpha$ y $p^t x \leq x$, $\forall x \in S$.

Propiedades

- Sea $S \subseteq R^n$ un conjunto convexo y cerrado. Entonces S es la intersección de todos los semiespacios que contienen a S.
- Sea $S \subseteq R^n$ un conjunto no vacío, y sea $y \notin Cl(env(S))$, la clausura de la envoltura de S. Entonces, existe un hiperplano que separa fuertemente a S y a y.

Soporte de conjuntos en puntos frontera

Sea $S \subseteq R^n$ un conjunto no vacío, y sea $\overline{x} \in Fr(S)$. Un hiperplano $H = \{x : p^t(x - \overline{x}) = 0\}$ es llamado *hiperplano soporte* de S en $\overline{x}$ si $p^t(x - \overline{x}) \geq 0$, $\forall x \in S$, o $p^t(x - \overline{x}) \leq 0$, $\forall x \in S$. Si además, $S \not\subset H$, H es *hiperplano soporte adecuado* de S en $\overline{x}$.

Teorema

Sea $S \subseteq R^n$ un conjunto no vacío convexo, y sea $\overline{x} \in Fr(S)$. Entonces, existe un *hiperplano soporte de* S en $\overline{x}$, es decir, $\exists p$ vector no nulo tal que $p^t(x - \overline{x}) \leq 0, \forall x \in Cl(S)$.

Propiedades

- Sea $S \subseteq R^n$ un conjunto convexo no vacío y $\overline{x} \notin Int(S)$. Entonces, $\exists p$ vector no nulo tal que $p^t(x - \overline{x}) \leq 0, \forall x \in Cl(S)$.
- Sea $S \subseteq R^n$ un conjunto convexo no vacío y $y \notin Int(Cl(S))$. Entonces, existe un hiperplano que separa S y a y.
- Sea $S \subseteq R^n$ un conjunto convexo no vacío y sea $\overline{x} \in Fr(x) \cap Fr(Cl(S))$. Entonces, existe un hiperplano de soporte de S a $\overline{x}$.

Separación de dos conjuntos convexos

Sea $S_1, S_2 \subseteq R^n$ conjuntos convexos no vacíos, y suponemos que $S_1 \cap S_2 = \emptyset$. Entonces, existe un hiperplano que *separa* S_1 y S_2. Es decir, $\exists p \in R^n$ vector no nulo tal que

$$inf\{p^t x : x \in S_1\} \geq sup\{p^t x : x \in S_2\}$$

1.2.6. Teoremas de la alternativa

Los Teoremas de la Alternativa establecen condiciones para la existencia de soluciones de sistemas de ecuaciones o inecuaciones lineales. Demuestran que, dados dos sistemas, si uno de ellos no tiene solución, el otro sí la tiene. Como ejemplo de estos teoremas, citamos [5]:

- **Teorema de Farkas.**
  O el sistema
$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

tiene solución, o el sistema

$$\begin{aligned} A^T y &\geq 0 \\ b^t \cdot y &< 0 \end{aligned}$$

tiene solución.

- O el sistema

$$Ax \leq b$$

tiene solución, o el sistema

$$\begin{aligned} A^T y &= 0 \\ y &\geq 0 \\ b \cdot y^t &< 0 \end{aligned}$$

tiene solución.

- O el sistema

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

tiene solución, o el sistema

$$\begin{aligned} A^T y &\geq 0 \\ y &\geq 0 \\ b \cdot y^t &< 0 \end{aligned}$$

tiene solución.

1.3. Funciones convexas

Si $S \subseteq R^n$ es un conjunto convexo, diremos que una función $f : S \to R$ es **convexa** sí y solo sí:

- $\forall x, y \in S$, y $\forall \theta \in [0, 1]$, tenemos:

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

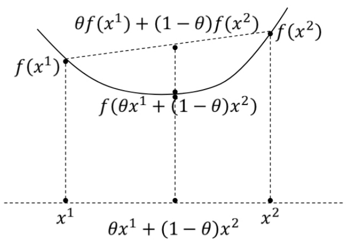


Figura 1.2: Representación gráfica de una función convexa

1.3.1. Funciones estrictamente convexas

Una función f es **estrictamente convexa** si la desigualdad anterior se cumple siempre que $x \neq y, \forall \theta \in [0,1]$, es decir:

$$f(\theta x + 1(1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

1.3.2. Funciones cóncavas

Si $S \subseteq R^n$ es un conjunto convexo, diremos que una función $f : S \to R$ es **cóncava** sí y solo sí $-f$ es convexa, es decir

$\forall x, y \in S$, y $\forall \theta \in [0,1]$, tenemos

$$f(\theta x + 1(1-\theta)y) \geq \theta f(x) + (1-\theta)f(y)$$

- f es estrictamente cóncava $\Leftrightarrow -f$ es estrictamente convexa.
- f es estrictamente convexa $\Rightarrow f$ es convexa pero no recíprocamente.

1.3.3. Subgradiente y propiedades

Dados $S \subseteq R^n$ convexo, $f : S \to R$ convexa y $\overline{x} \in S$, diremos que ξ es un **subgradiente** para f en $\overline{x}$ sí, y solo sí

$$f(x) \geq f(\overline{x}) + \xi^t(x-\overline{x}) \ \forall x \in S$$

Ejemplo

$$f(x) = \begin{cases} -2x+3 \ si \ x \leq 2 \\ \\ x^2-5 \quad si \ x > 2 \end{cases}$$

En $\overline{x} = 2$, el conjunto de subgradientes de f es el intervalo $[-2,4]$.

Se puede observar que el concepto de subgradiente establece la subdiferenciabilidad para las funciones convexas y cóncavas.

Propiedad

Sea $S \subseteq R^n$ un conjunto convexo no vacío y sea $f : S \to R$ una función. Suponemos que: $\forall \ \overline{x} \in int(S)$, $\exists \ \xi$ vector subgradiente tal que

$$f(x) \geq f(\overline{x}) + \xi'(x-\overline{x}), \forall \overline{x} \in S$$

entonces, f es convexa en $int(S)$.

El concepto de subgradiente permite dar una nueva caracterización de las funciones convexas:

Proposición

Sean S un conjunto convexo y abierto, $f : S \subseteq R^n \rightarrow R$. Entonces:
f es convexa $\Leftrightarrow \forall \overline{x} \in S$ existe ξ, subgradiente de f en $\overline{x}$ tal que

$$f(x) \geq f(\overline{x}) + \xi^t(x - \overline{x}), \forall x \in S$$

1.3.4. Diferenciabilidad y propiedades

Dados $f : S \subseteq R^n \rightarrow R$, $\overline{x} \in int(S)$, diremos que f es **diferenciable** en $\overline{x}$ si, y solo si, existe un vector $\nabla f(\overline{x})$, denominado gradiente de f en $\overline{x}$, y existe una función $\alpha : R^n \rightarrow R$ tal que

$$f(x) = f(\overline{x}) + \nabla f(\overline{x})^t(x - \overline{x}) + \|x - \overline{x}\|\alpha(\overline{x}; x - \overline{x})$$

con $\quad \lim_{x \rightarrow \overline{x}} \alpha(\overline{x}; x - \overline{x}) = 0$.

Diremos que f es diferenciable en $S' \subseteq S$, con S' abierto, si es diferenciable en cualquier punto $\overline{x} \in S'$.

Por otro lado, $\nabla f(\overline{x})^t = (\frac{\partial f(\overline{x})}{\partial x_1}, \cdots, \frac{\partial f(\overline{x})}{\partial x_n})$, es decir, el gradiente, es el vector de las derivadas parciales de f en $\overline{x}$.

Proposición

Si f es diferenciable en $\overline{x} \in int(S)$, entonces f es subdiferenciable en $\overline{x}$ y el conjunto de subgradientes de f en $\overline{x}$ es unitario y se reduce a $\nabla f(\overline{x})$.

Por tanto, en el caso diferenciable se puede dar la siguiente caracterización de convexidad:

Proposición

Sean $S \subseteq R^n$ convexo y abierto, $f : S \rightarrow R$, diferenciable en S:

$$f \text{ es convexa} \Leftrightarrow f(x) \geq f(\overline{x}) + \nabla f(\overline{x})^t(x - \overline{x}), \forall x \in S$$

1.3.5. Diferenciabilidad doble

Dados $f : S \subseteq R^n \rightarrow R, \overline{x} \in int(S)$, diremos que f es **diferenciable dos veces** en $\overline{x}$ sí y solo sí, existe un vector $\nabla f(\overline{x})$, denominado gradiente de f en $\overline{x}$, existe una matriz simétrica de orden $n \times n$, $H(\overline{x})$, denominada hesiano, y existe una función $\alpha : R^n \rightarrow R$ tales que

$$f(x) = f(\overline{x}) + \nabla f(\overline{x})^t(x - \overline{x}) + \frac{1}{2}(x - \overline{x})^t H(\overline{x})(x - \overline{x}) + \|x - \overline{x}\|^2\alpha(\overline{x}; x - \overline{x})$$

Con $\quad \lim_{x \rightarrow \overline{x}} \alpha(\overline{x}; x - \overline{x}) = 0$.

Diremos que f es dos veces diferenciable en $S' \subseteq S$ con S' abierto, si es dos veces diferenciable en cualquier punto $\overline{x} \in S'$. La matriz $H(\overline{x})$ contiene las segundas derivadas parciales de f, es decir: $h_{ij} = \frac{\partial^2 f(\overline{x})}{\partial x_i \partial x_j}$ para i,j=1, …, n.

Proposición

Sea $S \subseteq R^n$ convexo y abierto, $f : S \to R$, es diferenciable dos veces en S.

f es convexa $\Leftrightarrow \forall \overline{x} \in S, H(\overline{x})$ es una matriz semidefinida positiva.

1.4. Funciones convexas generalizadas

Como generalización de funciones convexas, introducimos funciones cuasiconvexas, estrictamente y fuertemente cuasiconvexas y pseudoconvexas y prestamos atención a algunas propiedades relacionadas con estos nuevos conceptos.

1.4.1. Funciones cuasiconvexas

Si $S \subseteq R^n$ es un conjunto convexo, diremos que una función $f : S \to R$ es *cuasiconvexa* si, y solo si

$$\forall x, y \in S, \forall \theta \in (0,1), \text{ se tiene que } f(\theta x + (1-\theta)y) \leq max\{f(x), f(y)\}$$

Además,

$$f \text{ cuasicóncava} \Leftrightarrow -f \text{ cuasiconvexa.}$$

Vemos que f convexa $\Rightarrow f$ cuasiconvexa, pero no se cumple el recíproco.

1.4.2. Funciones estrictamente cuasiconvexas

Si $S \subseteq R^n$ es un conjunto convexo, diremos que una función $f : S \to R$ es *estrictamente cuasiconvexa* si, y solo si

$$\forall x, y \in S,\ \forall \theta \in (0,1),\ f(x) \neq f(y)$$

se tiene que

$$f(\theta x + (1-\theta)y) < max\{f(x), f(y)\}$$

Además,

$$f \text{ estrictamente cuasicóncava} \Leftrightarrow -f \text{ estrictamente cuasiconvexa.}$$

También ocurre, como en el caso anterior, que

$$f \text{ convexa} \Rightarrow f \text{ estrictamente cuasiconvexa}$$

pero no se cumple el recíproco. Sin embargo, entre una función cuasiconvexa y una función estrictamente cuasiconvexa no existe esta relación, ya que una función estrictamente cuasiconvexa no es necesariamente cuasiconvexa ni recíprocamente.

Ejemplo

$$f(x) = \begin{cases} 1 \; si \; x = 0 \\ \\ 0 \; si \; x \neq 0 \end{cases}$$ es estrictamente cuasiconvexa, pero no cuasiconvexa.

Ahora, vamos a dar una definición que sí puede relacionar estos conceptos.

1.4.3. Funciones semicontinuas inferiormente

$f : S \subseteq R^n \to R$ es *semicontinua inferiormente* en $\overline{x} \in S$, si $\forall \varepsilon > 0, \ni \partial > 0$ tal que $\forall x \in S$ con $\|x - \overline{x}\| < \partial$ se tiene que

$$f(x) - f(\overline{x}) > -\varepsilon$$

Si f es semicontinua inferiormente en todos los puntos de S, entonces f es semicontinua sobre S.

Proposición

Si f es estrictamente cuasiconvexa y semicontinua inferiormente, entonces f es cuasiconvexa.

1.4.4. Funciones fuertemente cuasiconvexas

Si $S \subseteq R^n$ es un conjunto convexo, diremos que una función $f : S \to R$ es *fuertemente cuasiconvexa* si, y solo si

$$\forall x, y \in S, x \neq y, \forall \theta \in (0, 1)$$

se tiene que

$$f(\theta x + (1 - \theta)y) < max\{f(x), f(y)\},$$

Además,

$$f \text{ fuertemente cuasicóncava} \Leftrightarrow -f \text{ fuertemente cuasiconvexa.}$$

Viendo todas estas definiciones, podemos agrupar las propiedades siguientes.

Propiedades

- f estrictamente convexa $\to$ f fuertemente cuasiconvexa
- f fuertemente cuasiconvexa $\to$ f estrictamente cuasiconvexa
- f fuertemente cuasiconvexa $\to$ f cuasiconvexa

1.4.5. Funciones pseudoconvexas

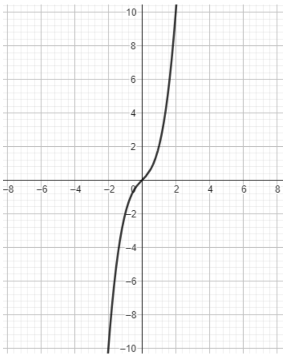
Sean $S \subseteq R^n$ abierto y $f : S \to R$ diferenciable en R. f es *pseudoconvexa* si, y solo si

$$\forall x, y \in S \text{ tales que } \nabla f(y)^t(x-y) \geq 0, \text{ se tiene que } f(x) \geq f(y)$$

Vemos que

$$f \text{ pseudocóncava} \Leftrightarrow -f \text{ pseudoconvexa.}$$

Ejemplos



- $f(x) = x + x^3$ es una función pseudoconvexa, pero no convexa (como vemos en la imagen).
- $f(x) = \frac{e^x}{x^2+1} + \frac{1}{e^x}$ es una función cuasiconvexa, pero no pseudoconvexa.

1.4.6. Funciones estrictamente pseudoconvexas

Sean $S \subseteq R^n$ abierto y $f : S \to R$ diferenciable en R. f es *estrictamente pseudoconvexa* si, y solo si

$$\forall x, y \in S,\ x \neq y, \text{ tales que } \nabla f(y)^t(x-y) \geq 0, \text{ se tiene que } f(x) > f(y)$$

$$f \text{ estrictamente pseudocóncava} \Leftrightarrow -f \text{ estrictamente pseudoconvexa.}$$

Proposición

Sean $S \subseteq R^n$ abierto y $f : S \to R$ diferenciable en R.

- f es pseudoconvexa $\to$ f es estrictamente cuasiconvexa y cuasiconvexa.
- f es estrictamente pseudoconvexa $\to$ f es fuertemente cuasiconvexa.

2

Problemas de optimización convexa

En este capítulo estudiamos los problemas de optimización convexa y sus propiedades. Extendemos este estudio al caso de problemas de optimización con funciones convexas generalizadas como paso previo a la presentación de las condiciones de optimalidad para problemas de programación no lineal.

2.1. Optimización convexa

Sea $f : S \to R$ y el problema

$$\begin{array}{c} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array} \tag{2.1}$$

donde $f(x)$ es una función convexa, y $S \subseteq R^n$, $S \neq \emptyset$ es un conjunto convexo. Con esta notación, describimos un problema donde buscamos un determinado punto x que minimiza $f(x)$, ttal que satisface las condiciones de S. Al conjunto S se le llama región factible, que viene dada por la forma $S = \{x \in R^n : Ax = b, x \geq 0\}$.

Llamamos a $x \in R^n$ la variable a optimizar, y a la función $f : R^n \to R$ la función objetivo. Se dice que el problema es factible si existe al menos un punto factible, y no factible, de otra manera.

2.1.1. Ejemplos

- Consideramos, por ejemplo, el problema

$$\begin{array}{rl} \text{mín} & f(x) = x^2 + y^2 \\ \text{s.a:} & g_1(x) = 4x \leq 0 \\ & g_2(x) = x + y = 0 \end{array}$$

  Vemos que este problema es convexo, ya que la función a optimizar es convexa, y las restricciones forman un conjunto convexo.

- Programacón cuadrática:

$$\begin{array}{l} \text{mín } \frac{1}{2}x^T Hx + c^T x \\ \text{s.a: } Ax \leq B \end{array}$$

con H definida positiva ($x^t Hx > 0$, $\forall x \in R^n$)

2.1.2. Conceptos de solución

- Si $\overline{x} \in S$ y un $N_\varepsilon(\overline{x})$, entorno centrado en $\overline{x}$ y de radio ε tal que $f(x) \geq f(\overline{x}), \forall x \in S \cap N_\varepsilon(\overline{x})$, entonces $\overline{x}$ es solución óptima local.
- De igual manera, si $\overline{x} \in S$ y $f(x) > f(\overline{x}), \forall x \in S \cap N_\varepsilon(\overline{x}), x \neq \overline{x}$, para algún $\varepsilon > 0$, entonces $\overline{x}$ es una solución óptima local estricta.
- Por otro lado, si $\overline{x} \in S$ es el único mínimo local en $S \cap N_\varepsilon(\overline{x})$, para un cierto $N_\varepsilon(\overline{x})$ alrededor de $\overline{x}$, entonces $\overline{x}$ es denominada solución óptima local fuerte.

- Además, si $\overline{x} \in S$ y $f(x) \geq f(\overline{x}), \forall x \in S$, entonces $\overline{x}$ es una solución óptima global.

2.2. Condiciones de optimalidad

Proposición

Sean $S \subseteq R^n$ convexo y $f : S \to R$ convexa. Supongamos que $\overline{x} \in S$ es una solución óptima local de 2.1. Entonces:

- $\overline{x}$ es una solución óptima global.
- Si $\overline{x}$ es una solución óptima global fuerte o f es estrictamente convexa, entonces $\overline{x}$ es solución óptima global única.

Proposición

Sean $S \subseteq R^n$ convexo y $f : S \to R$ convexa, y además consideramos el problema 2.1.

- $\overline{x}$ es una solución óptima del problema anterior si, y solo si, $\exists \xi$ subgradiente de f en $\overline{x}$ tal que

$$\xi^t (x - \overline{x}) \leq 0, \forall x \in S$$

Corolario 1

Sean $S \subseteq R^n$ convexo y $f : S \subseteq R^n \to R$ convexa, y además S abierto. Si consideramos el problema 2.1.

- $\overline{x} \in S$ es solución óptima de este problema si, y solo si, $\xi = 0$ es un subgradiente de f en $\overline{x}$.

Corolario 2

Sean $S \subseteq R^n$ convexo y $f : S \to R$ convexa, y además S abierto y f diferenciable. Si consideramos el problema 2.1.

- $\overline{x} \in S$ es solución óptima del problema anterior si, y solo si, $\nabla f(\overline{x}) = 0$.

2.3. Resultados de optimalidad para funciones convexas generalizadas

Veamos algunos resutados de optimalidad para funciones generalizadas, sacados de la sección 4.6 de [3].

Proposición

Sean $S \subseteq R^n$ un poliedro compacto (es decir, cerrado y acotado), $f : S \to R$ una función cuasiconvexa y el problema

$$\begin{array}{c} \text{máx } f(x) \\ \text{s.a: } x \in S \end{array}$$

entonces, existe una solución óptima global de este problema y ésta se alcanza en un punto extremo de S.

Proposición

Sean $S \subseteq R^n$ convexo y $f : S \to R$ estrictamente cuasiconvexa. Si tenemos que $\overline{x} \in S$ es una solución óptima local de

$$\begin{array}{c} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array}$$

entonces, $\overline{x} \in S$ es solución óptima global de dicho problema.

Proposición

Sean $S \subseteq R^n$ convexo y $f : S \to R$ fuertemente cuasiconvexa. Si tenemos que $\overline{x} \in S$ es una solución óptima local de

$$\begin{array}{c} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array}$$

entonces, $\overline{x} \in S$ es solución óptima global de dicho problema.

Proposición

Sean $S \subseteq R^n$ abierto y convexo, y $f : S \to R$ diferenciable en R y pseudoconvexa. Dado el problema

$$\begin{array}{c} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array}$$

entonces, $\overline{x} \in S$ es solución óptima global de dicho problema si, y solo si $\nabla f(\overline{x}) = 0$.

2.4. Condiciones de optimalidad en problemas de Programación no Lineal

Como ya hemos mencionado antes, los problemas de optimización convexa son un caso especial de los problemas de programación no lineal. La diferencia más fundamental entre los problemas de optimización convexa y los problemas de programación no lineal no convexos es que, para los primeros, se garantiza que los óptimos locales son globales. Las condiciones de optimalidad que vamos a ver a continuación se establecen para problemas de optimización no lineal con las propiedades de convexidad y diferenciabilidad adecuadas.

Teorema (Condición geométrica)

Consideramos el problema de optimización

$$\begin{array}{c} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array}$$

con $S = \{x \in X : g_i(x) \leq 0, i = 1, \ldots, n\} \subseteq R^n$, donde X es un conjunto abierto no vacío en R^n. Dado un punto factible $\overline{x}$, sea $I = \{i : g_i(\overline{x}) = 0$ el conjunto de índices para las restricciones que se dan con igualdad en $\overline{x}$. Además, suponemos que f y g_i para $i \in I$ son diferenciables en $\overline{x}$ y que g_i para $i \notin I$ son continuas en $\overline{x}$.

Si $\overline{x}$ en una solucion óptima local, entonces $F_0 \cap G_0 = \emptyset$, donde

$$F_0 = \{d : \nabla f(\overline{x})^t d < 0\} \text{ y } G_0 = \{d : \nabla g_i(\overline{x})^t d < 0 \text{ para cada } i \in I\}.$$

Inversamente, si $F_0 \cap G_0 = \emptyset$, f es pseudoconvexa en $\overline{x}$ y $g_i, i \in I$ son estrictamente pseudoconvexas en algun entorno ϵ de $\overline{x}$, entonces $\overline{x}$ es un mínimo local.

2.4.1. Condiciones de optimalidad de Fritz John

Ahora extendemos la condición de optimalidad geométrica necesaria de $F_0 \cap G_0 = \emptyset$ a una declaración en términos de los gradientes de la función objetivo y de las restricciones vinculantes (aquellas g_i que cumplen $i \in I$). Las condiciones de optimización resultantes son las condiciones de optimalidad de Fritz John.

Teorema (Condiciones necesarias de Fritz John)

Sea X un conjunto abierto no vacío en R^n y sea $f : R^n \to R$, y $g_i : R^n \to R, i = 1, \ldots, m$. Consideramos el problema de optimización

$$\begin{aligned} &\text{mín } f(x) \\ &\text{s.a: } g_i(x) \leq 0, i = 1, \ldots, n \\ &\quad\; x \in X \end{aligned}$$

Sea $\overline{x}$ una solución factible, y denotamos $I = \{i : g_i(\overline{x}) = 0\}$. Además, suponemos que f y g_i para $i \in I$ son diferenciables en $\overline{x}$ y que g_i para $i \notin I$ son continuas en $\overline{x}$. Si $\overline{x}$ en una solución óptima local, existen unos escalares u_0 y u_i para $i \in I$ tal que

$$u_0 \nabla f(x) + \textstyle\sum_{i \in I} u_i \nabla g_i(\overline{x}) = 0$$

$$u_0, u_i \geq 0$$

Además, si g_i para $i \notin I$ son también diferenciables en $\overline{x}$, las condiciones anteriores se pueden escribir en la siguiente forma equivalente:

$$u_0 \nabla f(x) + \textstyle\sum_{i \in I}^{m} u_i \nabla g_i(\overline{x}) = 0$$

$$u_i g_i(\overline{x}) = 0$$

$$u_0, u_i \geq 0$$

Teorema (Condiciones suficientes de Fritz John)

Sea X un conjunto abierto no vacío en R^n y sea $f : R^n \to R$, y $g_i : R^n \to R, i = 1, \ldots, m$. Consideramos el problema de optimización

$$\begin{aligned} &\text{mín } f(x) \\ &\text{s.a: } g_i(x) \leq 0, i = 1, \ldots, n \\ &\qquad x \in X \end{aligned}$$

Sea $\overline{x}$ una solución FJ, y denotamos $I = \{i : g_i(\overline{x}) = 0\}$. Definimos S como la región factible relajada del problema en el que se eliminan las restricciones no vinculantes, es decir, aquellas $g_i(x)$ que no cumplen $i \in I$.

- Si existe un entorno $N_\varepsilon(\overline{x})$, $\varepsilon > 0$, tal que f es pseudoconvexa sobre $S \cap N_\varepsilon(\overline{x})$, y $g_i, i \in I$ es estrictamente pseudoconvexa sobre $S \cap N_\varepsilon(\overline{x})$, entonces $\overline{x}$ es un mínimo local para el problema.
- Si f es pseudoconvexa en $\overline{x}$, y si $g_i, i \in I$ son estrictamente pseudoconvexa y cuasiconvexa en $\overline{x}$, $\overline{x}$ es una solución global óptima para el problema. En particular, si estos supuestos de convexidad generalizada se mantienen restringiendo el dominio e f en $N_\varepsilon(\overline{x})$ para algún $\varepsilon > 0$, entonces $\overline{x}$ es un mínimo local para el problema.

2.4.2. Condiciones de optimalidad de Karush Kuhnn Tucker

Teorema (Condiciones necesarias de Karush Kuhnn Tucker)

Sea X un conjunto abierto no vacío en R^n y sea $f : R^n \to R$, y $g_i : R^n \to R, i = 1, \ldots, m$. Consideramos el problema de optimización

$$\begin{aligned} &\text{mín } f(x) \\ &\text{s.a: } g_i(x) \leq 0, i = 1, \ldots, n \\ &\qquad x \in X \end{aligned}$$

Sea $\overline{x}$ la solución factible, y denotamos $I = \{i : g_i(\overline{x}) = 0\}$. Suponemos que f y g_i para $i \in I$ son son diferenciables en $\overline{x}$ y que g_i para $i \notin I$ son continuas en $\overline{x}$. Además, suponemos que $\nabla g_i(\overline{x})$ para $i \in I$ son linealmente independientes. Si $\overline{x}$ es la solución local del problema, existen escalares u_i para $i \in I$ tal que

$$u_0 \nabla f(x) + \textstyle\sum_{i \in I} u_i \nabla g_i(\overline{x}) = 0$$

$$u_i \geq 0, \text{ para } i \in I$$

Además de los supuestos anteriores, si g_i para cada $i \notin I$ es también differenciable en $\overline{x}$, las anteriores condiciones pueden escribirse de la siguiente forma:

$$u_0 \nabla f(x) + \textstyle\sum_{i \in I}^{m} u_i \nabla g_i(\overline{x}) = 0$$

$$u_i g_i(\overline{x}) = 0, \text{ para } i = 1, \ldots, n$$

$$u_i \geq 0, \text{ para } i = 0, \ldots, n$$

Teorema (Condiciones de Karush Kuhnn Tucker y aproximaciones de PL de primer orden)

Sea X un conjunto abierto no vacío en R^n y sea $f : R^n \to R$, y $g_i : R^n \to R, i = 1, \ldots, n$ funciones diferenciables. Consideramos el problema de optimización

$$\begin{array}{l} \text{mín } f(x) \\ \text{s.a: } x \in S \end{array}$$

con $S = \{x \in X : g_i(x) \leq 0, i = 1, \ldots, n\}$. Dado un punto factible $\overline{x}$, y denotamos $I = \{i : g_i(\overline{x}) = 0\}$.

Definimos

$$F_0 = \{d : \nabla f(\overline{x})^t d < 0\} \text{ y } G'_0 = \{d \neq 0 : \nabla g_i(\overline{x})^t d \leq 0, \text{ para cada } i \in I\}$$

y sea

$$G' = \{d : \nabla g_i(\overline{x})^t d \leq 0, \text{ para cada } i \in I\} = G'_0 \cup \{0\}$$

Entonces, $\overline{x}$ es una KKT solución sí y solo sí $F_0 \cap G' = \emptyset$, que es equivalente a $F_0 \cap G'_0 = \emptyset$. Además, consideramos la aproximación de PL de primer orden del problema:

$$\text{LP}(\overline{x}): Min\{f(\overline{x}) + \nabla f(\overline{x})^t(x - \overline{x}) : g_i(\overline{x}) + \nabla g_i(\overline{x})^t(x - \overline{x}) \leq 0\} \text{ para } i = 1, \ldots, n$$

Entonces, $\overline{x}$ es una solución KKT sí y solo sí $\overline{x}$ soluciona LP($\overline{x}$).

Teorema(Condiciones suficientes de Karush Kuhnn Tucker)

Sea X un conjunto abierto no vacío en R^n y sea $f : R^n \subseteq R$, y $g_i : R^n \subseteq R, i = 1, \ldots, m$. Consideramos el problema de optimización

$$\begin{array}{l} \text{mín } f(x) \\ \text{s.a: } g_i(x) \leq 0, i = 1, \ldots, n \end{array}$$

Sea $\overline{x}$ una solución KKT, y denotamos $I = \{i : g_i(\overline{x}) = 0\}$. Definimos S como la región factible relajada del problema en el que se eliminan las restricciones no vinculantes. Entonces:

- Si existe un entorno $N_\varepsilon(\overline{x})$ sobre $\overline{x}$,$\varepsilon > 0$, tal que f es pseudoconvexa sobre $S \cap N_\varepsilon(\overline{x})$, y g_i son diferenciables sobre $\overline{x}$ para $i \in I$ y cuasiconvexa sobre $S \cap N_\varepsilon(\overline{x})$, entonces $\overline{x}$ es un mínimo local para el problema.
- Si f es pseudoconvexa en $\overline{x}$, y si $g_i, i \in I$ son diferenciables y cuasiconvexas, $\overline{x}$ es una solución global óptima para el problema. En particular, si estos supuestos de convexidad generalizada se mantienen restringiendo el dominio de la restricción factible en $N_\varepsilon(\overline{x})$ para algún $\varepsilon > 0$, entonces $\overline{x}$ es un mínimo local para el problema.

3

Métodos de resolución de problemas de programación convexa

3.1. Introducción

En este capítulo exploramos tres métodos que pueden ser utilizados para la resolución de algunos problemas de programación convexa. Los métodos que examinamos son el método del Gradiente Reducido [1], el método de Simplex Convexo de Zangwill [1] y el método de Dirección Alterna de Multiplicadores [2]. Cada uno de estos métodos aborda los problemas de programación convexa desde una perspectiva diferente y ofrece ventajas y desventajas únicas en términos de convergencia, velocidad de ejecución y complejidad algorítmica.

3.2. Método del Gradiente Reducido

El método del Gradiente Reducido [1] es un algoritmo de optimización no lineal utilizado para minimizar una función, en general, diferenciable sobre un poliedro. Este método se basa en un algoritmo que se mueve en la dirección opuesta del gradiente. El gradiente reducido utiliza una versión simplificada del gradiente, tal y como su nombre indica, iterando sobre el problema para actualizar la solución candidata.

El método del Gradiente Reducido es una técnica particularmente efectiva para problemas de programación convexa, ya que la convexidad de la función garantiza que cualquier mínimo local sea un mínimo global, lo que permite que el método encuentre una solución óptima. Si además la función objetivo es escrictamente convexa, nos aseguramos de que este mínimo global sea único. En caso de que la función sea diferenciable, también podemos afirmar que su gradiente siempre apunta en la dirección de mayor crecimiento de la función.

En el ejemplo que veremos más adelante, aplicaremos el método del Gradiente Reducido a un problema de programación convexa. Veremos cómo se puede implementar el algoritmo de gradiente reducido para encontrar la solución óptima del problema, así como utilizar el gradiente reducido para actualizar la solución candidata en cada iteración.

Consideramos el siguiente problema:

$$\begin{aligned} &\text{mín } f(x) \\ &\text{s.a: } Ax = b \\ &\qquad x \geq 0 \end{aligned} \tag{3.1}$$

donde A es una matriz $m \times n$ de rango m, $b \in R^m$ es un vector, y f una función convexa y diferenciable con continuidad en R^n. Supondremos que cada punto extremo de la región factible tiene m variables estrictamente positivas. Con este supuesto, cada solución factible tiene como mínimo m componentes positivas y, como máximo, $n - m$ componentes cero.

Sea x una solución factible. Por la anterior suposición, podemos descomponer A en $[B, N]$, y x^t en $[x_B^t, x_N^t]$, donde B es una matriz invertible $m \times m$, y $x_B > 0$. x_B es el vector básico y cada una de sus componentes es estrictamente positiva. Las componentes del vector no básico x_N pueden ser tanto positivas como cero.

Sea

$$\nabla f(x)^t = [\nabla_B f(x)^t, \nabla_N f(x)^t] \tag{3.2}$$

donde $\nabla_B f(x)^t$ es el gradiente de f respecto al vector x_B, y $\nabla_N f(x)^t$ es el gradiente de f respecto al vector x_N.

Decimos que d es una **dirección factible de mejora** de f en x si se cumple $\nabla f(x)^t d < 0$, y si $Ad = 0$ con $d_j \geq 0$ si $x_j = 0$. Nos interesa determinar un vector de dirección d que satisfaga estas propiedades. Primero, descomponemos d^t en $[d_B^t, d_N^t]$. Tengamos en cuenta que $0 = Ad = Bd_B = Nd_N$ se cumple automáticamente si $\forall d_N$, $d_B = -B^{-1}Nd_N$. Sea

$$r^t = (r_B^t, r_N^t) = \nabla f(x)^t - \nabla_B f(x)^t B^{-1} A = [0, \nabla_N f(x)^t - \nabla_B f(x) B^{-1} N] \tag{3.3}$$

el **gradiente reducido**. Tenemos que:

$$\nabla f(x)^t d = \nabla_B f(x)^t d_B + \nabla_N f(x)^t d_N = [\nabla_N f(x)^t - \nabla_B f(x)^t B^{-1} N] d_N = r_N^t d_N \tag{3.4}$$

Debemos elegir d_N de manera que $r_N^t d_N < 0$ y que $d_j \geq 0$ si $x_j = 0$.

Se adopta la siguiente regla. Para cada componente no básica j, se cumple $d_j = -r_j$ si $r_j \leq 0$ y $d_j = -x_j r_j$ si $r_j > 0$. Esto asegura que $d_j \geq 0$ si $x_j = 0$, y evita pasos indebidamente pequeños cuando $x > 0$ es muy pequeño, mientras $r_j > 0$. Esto también ayuda a la convergencia.

Para resumir, hemos descrito un procedimiento para construir una dirección factible de mejora.

3.2.1. Resumen de algoritmo del Gradiente Reducido

A continuación, vamos a resumir el algoritmo del Gradiente Reducido[1] para resolver un problema de minimizar $f(x)$ sujeto a $Ax = b, x \geq 0$. Se supone que todas las m columnas de A son linealmente independientes y que cada punto extremo de la región factible tiene m componentes estrictamente positivas. Como mostraremos en breve, el algoritmo converge a un punto **KKT** (visto en 2.4), siempre que las variables básicas se elijan correctamente.

Paso inicial

Elegimos un punto x^1 que satisfaga $Ax^1 = b$, $x^1 \geq 0$. Sea $k = 1$ y comencemos el paso principal.

Paso principal

1. Sea $d_k^t = (d_B^t, d_N^t)$ donde d_N y d_B se obtienen como sigue a partir de 3.8 y 3.9, respectivamente. Si $d = 0$, paramos: x es un punto **KKT**. (Los multiplicadores de Lagrange asociados con $Ax = b$ y $x \geq 0$ son, respectivamente, $\nabla_B f(x^k)^t B^{-1}$ y r) Si esto no se cumple, pasamos al paso 2.
Sea

$$I_k = \text{Conjunto de índices de las } m \text{ componentes más grandes de } x_k \tag{3.5}$$

$$B = \{a_j : j \in I_k\}, N = \{a_j : j \notin I_k\} \tag{3.6}$$

$$r^t = \nabla f(x^k)^t - \nabla_B f(x^k)^t B^{-1} A \tag{3.7}$$

$$d_j = \begin{cases} -r_j & si\ j \notin I_k, r_j \leq 0 \\ \\ -x_j r_j & si\ j \notin I_k, r_j > 0 \end{cases} \tag{3.8}$$

$$d_B = -B^{-1} N d_N \tag{3.9}$$

2. Resolvemos el siguiente problema de búsqueda escalar:

$$\begin{array}{l} \text{mín } f(x^k + \lambda^k) \\ \text{s.a: } 0 \leq \lambda \leq \lambda_{max} \end{array}$$

donde

$$\lambda_{max} = \begin{cases} min_{1 \leq j \leq n}\{\frac{-x_{jk}}{d_{jk}} : d_{jk} < 0\} & si\ d_k < 0 \\ \\ \infty & si\ d_k \geq 0 \end{cases} \tag{3.10}$$

y x_{jk}, d_{jk} son las $j-$ésimas componentes de x^k y d^k, respectivamente. Sea λ^k una solución óptima, y sea $x^{k+1} = \lambda^k d^k$. Reemplazamos k por $k = 1$ y volvemos al paso 1.

3.2.2. Ejemplo

Consideramos el siguiente problema convexo, donde hemos convertido las restricciones a igualdades con variables de holgura:

$$\begin{aligned} &\text{mín } 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ &\text{s.a: } x_1 + x_2 + x_3 = 2 \\ &\qquad x_1 + 5x_2 + x_4 = 5 \\ &\qquad x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Resolvemos este problema utilizando el método del gradiente reducido a partir del punto $x^1 = (0,0,2,5)'$. Debemos tener en cuenta que

$$\nabla f(x) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6, 0, 0)^t$$

Iteración 1:

Dirección de búsqueda: En el punto $x^1 = (0,0,2,5)^t$, tenemos $\nabla f(x^1) = (-4,-6,0,0)^t$. Por 3.5, tenemos $I_1 = \{3,4\}$, de modo que $B = [a_3, a_4]$ y $N = [a_1, a_2]$. Por 3.7, el gradiente reducido viene dado por

$$r^t = (-4,-6,0,0) - (0,0)\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix} = (-4,-6,0,0)$$

La información en este punto de la resolución del problema la podemos resumir aquí:

			x_1	x_2	x_3	x_4
Solución x^1			0	0	2	5
$\nabla f(x^1)$			-4	-6	0	0
$\nabla_B f(x^1) =$	0	x_3	1	1	1	0
	0	x_4	1	5	0	1
r			-4	-6	0	0

Tabla 3.1: Tabla asociada a x^1

Por 3.8, $d_N = (d_1, d_2)^t = (4,6)^t$. Calculamos d_B usando 3.9: $d_B = (d_3, d_4)^t = -B^{-1}Nd_N = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}\begin{pmatrix} 4 \\ 6 \end{pmatrix} = (-10,-34)^t$. Por tanto, el vector director es $d^1 = (4,6,-10,-34)^t$.

Movimiento en la dirección de búsqueda: A partir de $x^1 = (0,0,2,5)^t$, queremos minimizar la función objetivo en la direción $d^1 = (4,6,-10,-34)^t$. El valor máximo de λ tal que $x^1 + \lambda d^1$ sea factible, se calcula usando 3.10, y obtenemos: $\lambda_{max} = \text{mín}\{\frac{2}{10}, \frac{5}{34}\} = \frac{5}{34}$, y por tanto, $x^2 = x^1 + \lambda d^1 = (\frac{10}{17}, \frac{15}{17}, \frac{9}{17}, 0)^t$.

Iteración 2:

Dirección de búsqueda: En el punto $x^2 = (\frac{10}{17}, \frac{15}{17}, \frac{9}{17}, 0)^t$, por 3.5, tenemos $I_2 = \{1,2\}$, $B = [a_1, a_2]$ y $N = [a_3, a_4]$. Además, también tenemos que $\nabla f(x^2) = (-\frac{58}{17}, -\frac{62}{17}, 0, 0)^t$.

La información en este punto de la resolución del problema la podemos resumir aquí, donde las filas de x_1 y x_2 se obtienen mediante dos operaciones pivotaje en la tabla 3.1:

		x_1	x_2	x_3	x_4
Solución x^2		$\frac{10}{17}$	$\frac{15}{17}$	$\frac{9}{17}$	0
$\nabla f(x^2)$		$-\frac{58}{17}$	$-\frac{62}{17}$	0	0
$\nabla_B f(x^2) = \begin{bmatrix} -\frac{58}{17} \\ -\frac{62}{17} \end{bmatrix}$	x_1	1	0	$\frac{5}{4}$	$-\frac{1}{4}$
	x_2	0	1	$-\frac{1}{4}$	$\frac{1}{4}$
r		0	0	$\frac{57}{17}$	$\frac{1}{17}$

Tabla 3.2: Tabla asociada a x^2

Por 3.7, el gradiente reducido viene dado por

$$r^t = (-\frac{58}{17}, -\frac{62}{17}, 0, 0) - (-\frac{58}{17}, -\frac{62}{17}) \begin{bmatrix} 1 & 0 & \frac{5}{4} & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} = (0, 0, \frac{57}{17}, \frac{1}{17})$$

Por 3.8, $d_3 = -\frac{9}{17}\frac{57}{17} = -\frac{513}{289}$ y $d_4 = 0$, por lo que $d_N = (-\frac{513}{289}, 0)^t$. De 3.9 tenemos $d_B = (d_1, d_2)^t = \begin{bmatrix} \frac{5}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{pmatrix} -\frac{513}{289} \\ 0 \end{pmatrix} = (\frac{2565}{1156}, -\frac{513}{1156})^t$.

Por tanto, la nueva dirección de búsqueda viene dada por $d^2 = (\frac{2565}{1156}, -\frac{513}{1156}, -\frac{513}{289}, 0)^t$.

Movimiento en la dirección de búsqueda: A partir de $x^2 = (\frac{10}{17}, \frac{15}{17}, \frac{9}{17}, 0)^t$, queremos minimizar la función objetivo en la dirección $d^2 = (\frac{2565}{1156}, -\frac{513}{1156}, -\frac{513}{289}, 0)^t$. El valor máximo de λ tal que $x^2 + \lambda d^2$ sea factible, se calcula usando 3.10, y obtenemos: $\lambda_{max} = \frac{17}{57}$. Por tanto, $x^3 = x^2 + \lambda d^2 = (\frac{35}{31}, \frac{24}{31}, \frac{3}{31}, 0)^t$.

Iteración 3:

Dirección de búsqueda: Tenemos que $I_3 = \{1,2\}$, $B = [a_1, a_2]$ y $N = [a_3, a_4]$. Como $I_3 = I_2$, la tabla 3.2 puede ser conservada. En este momento, $\nabla f(x^3) = (-\frac{32}{31}, -\frac{160}{31}, 0, 0)^t$.

Por 3.7, el gradiente reducido viene por

$$r^t = (-\frac{32}{31}, -\frac{160}{31}, 0, 0) - (-\frac{32}{31}, -\frac{160}{31}) \begin{bmatrix} 1 & 0 & \frac{5}{4} & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} = (0, 0, 0, \frac{32}{31})$$

Por 3.8, $d_N = (d_3, d_4)^t = (0, 0)^t$. De 3.9 tenemos que $d_B = (d_1, d_2)^t = (0, 0)^t$. Por ello, $d = 0$, y la solución x^3 es una solución **KKT** y, por lo tanto, óptima para este problema.

	x_1	x_2	x_3	x_4
Solución x^3	$\frac{35}{31}$	$\frac{24}{31}$	$\frac{3}{31}$	0
$\nabla f(x^3)$	$-\frac{32}{31}$	$-\frac{160}{31}$	0	0
$\nabla_B f(x^3) = \begin{bmatrix} -\frac{32}{31} \\ -\frac{160}{31} \end{bmatrix}$ x_1	1	0	$\frac{5}{4}$	$-\frac{1}{4}$
x_2	0	1	$-\frac{1}{4}$	$\frac{1}{4}$
r	0	0	0	$\frac{32}{31}$

Tabla 3.3: Tabla asociada a x^3

Podemos ver gráficamente el proceso del algoritmo en la figura 3.1.

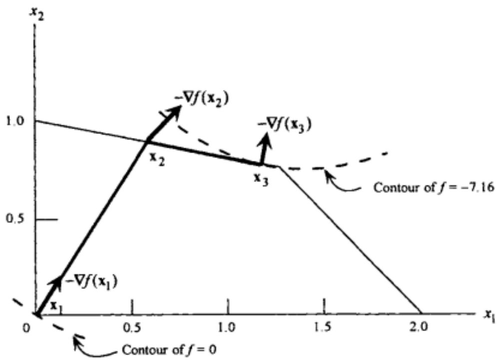


Figura 3.1: Proceso del método del Gradiente Reducido

3.2.3. Convergencia

El siguiente teorema contiene la convergencia del método del Gradiente Reducido a un punto **KKT**. Este teorema se explica más en detalle en la sección 10.6 de [1].

Teorema

Sea $f : R^n \to R$ una función continuamente diferenciable, y consideramos un problema de la forma 3.1.

A es una matriz $m \times n$ y $b \in R^m$ un vector tal que todos los puntos extremos de la región factible tiene m componentes positivas, y cualquier conjunto de las $m-$columnas de A son linealmente independientes. Suponemos que $\{x_k\}$ está generado por el algoritmo del Gradiente Reducido. Entonces, cualquier punto de acumulación de $\{x_k\}$ es un punto **KKT**.

Cabe destacar que hay dos suposiciones cruciales: la elección de las variables independientes y la de la dirección de búsqueda. Además, este teorema no proporciona información sobre la velocidad de convergencia, que puede variar dependiendo de las características específicas del problema de optimización. Sin embargo, si la función es convexa y no lineal, el mínimo de la función se va a alcanzar y va a ser único, con lo cual el método convergerá irremediablemente hacia él.

3.3. Método Simplex Convexo de Zangwill

El método simplex convexo de Zangwill[1] es similar al método de gradiente reducido de la sección anterior, a excepción de que en este se modifica una variable no básica, mientras que todas las demás variables no básicas se fijan en sus niveles actuales. Por supuesto, los valores de las variables básicas se modifican en consecuencia para mantener la factibilidad, de modo que el método se comporta de manera muy similar al método Simplex para problemas lineales. A continuación, reconstruimos este algortimo como una modificación del método de gradiente reducido para resolver problemas del tipo

$$\begin{aligned} &\text{mín } f(x) \\ &\text{s.a: } Ax = b \\ &\quad\;\; x \geq 0 \end{aligned} \tag{3.11}$$

donde f A es una matriz $m \times n$ de rango m, $b \in R^m$ es un vector y f una función convexa y diferenciable con continuidad en R^n.

3.3.1. Resumen del método Simplex Convexo

Suponemos de nuevo que cualesquiera m columnas de A son linealmente independientes y que todo punto extremo de la región factible tiene m componentes estrictamente positivas. Como mostraremos en breve, el algoritmo converge a un punto **KKT** (visto en 2.4), siempre que las variables básicas sean las m variables más positivas, donde un empate se rompe arbitrariamente.

Paso inicial

Elegimos un punto x^1 que satisfaga $Ax^1 = b$, $x^1 \geq 0$. Sea $k = 1$ y comenzamos el paso principal.

Paso principal

1. Dado x^k, identificamos I_k, B, N y calculamos r como:

$$I_k = \text{Conjunto de índices de las } m \text{ componentes más grandes de } x_k \tag{3.12}$$

$$B = \{a_j : j \in I_k\}, N = \{a_j : j \notin I_k\} \tag{3.13}$$

$$r^t = \nabla f(x^k)^t - \nabla_B f(x^k)^t B^{-1} A \tag{3.14}$$

Consideramos 3.15-3.20. Si $\alpha = \beta = 0$ (valores definidos posteriormente), paramos; x^k es un punto **KKT** que tiene multiplicadores de Lagrange $\nabla_B f(x^k)^t B^{-1}$ y r, respectivamente, asociados a las restricciones $Ax = b$ y $x \geq 0$. Si $\alpha > \beta$, calculamos d_N a partir de 3.17 y 3.18. Si $\alpha < \beta$, calculamos d_N a partir de 3.17 y 3.19. Si $\alpha \neq \beta$, calculamos d_N a partir de cualquiera de las dos opciones. En todos los casos, hallamos d_B de 3.20 y vamos al paso 2.

$$\alpha = \max\{-r_j : r_j \leq 0\} \tag{3.15}$$

$$\beta = \max\{x_j r_j : r_j \geq 0\} \tag{3.16}$$

$$\nu = \begin{cases} \text{un índice tal que } \alpha = -r_v \; si \; \alpha \geq \beta \\ \\ \text{un índice tal que } \alpha = x^v r_v \; si \; \alpha \leq \beta \end{cases} \tag{3.17}$$

$$\text{En caso de que se cumpla } \alpha \geq \beta : d_j = \begin{cases} 0 \; si \; j \notin I_k, j \neq v \\ \\ 1 \; si \; j \notin I_k, j = v \end{cases} \tag{3.18}$$

$$\text{En caso de que se cumpla } \alpha \leq \beta : d_j = \begin{cases} 0 \quad si \; j \notin I_k, j \neq v \\ \\ -1 \; si \; j \notin I_k, j = v \end{cases} \tag{3.19}$$

$$d_B = -B^{-1} N d_N = -B^{-1} a_v d_v \tag{3.20}$$

2. Resolvemos el siguiente problema escalar:

$$\begin{array}{l} \min f(x^k + \lambda d^k) \\ \text{s.a: } 0 \leq \lambda \leq \lambda_{max} \end{array}$$

donde

$$\lambda_{max} = \begin{cases} min_{1 \leq j \leq n}\{\frac{-x_{jk}}{d_{jk}} : d_{jk} < 0\} \; si \; d_k < 0 \\ \\ \infty \qquad\qquad\qquad\qquad\qquad si \; d_k \geq 0 \end{cases} \tag{3.21}$$

y x_{jk}, d_{jk} son las $j-$ésimas componentes de x_k y d_k, respectivamente. Sea λ_k una solución óptima, y sea $x_{k=1} x_k = \lambda_k d_k$. Reemplazamos k por $k = 1$ y volvemos al paso 1.

3.3.2. Ejemplo

Consideramos el siguiente problema:

$$\begin{aligned} &\text{mín } 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ &\text{s.a: } x_1 + x_2 + x_3 = 2 \\ &\qquad x_1 + 5x_2 + x_4 = 5 \\ &\qquad x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Resolvemos este problema utilizando el método Simplex Convexo de Zangwill a partir del punto $x^1 = (0,0,2,5)'$. Debemos tener en cuenta que:

$$\nabla f(x) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6, 0, 0)^t$$

Al igual que en el método de gradiente reducido, es conveniente exhibir la información en cada iteración en forma de tabla, dando el vector solución x^k y $\nabla f(x^k)$.

Iteración 1:

Dirección de búsqueda: En el punto $x^1 = (0,0,2,5)^t$, tenemos $\nabla f(x^1) = (-4,-6,0,0)^t$. Por 3.12, tenemos $I_1 = \{3,4\}$, de modo que $B = [a_3, a_4]$ y $N = [a_1, a_2]$. El gradiente reducido, usando 3.14, viene dado por

$$r^t = (-4,-6,0,0) - (0,0)\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{bmatrix} = (-4,-6,0,0)$$

La información en este punto de la resolución del problema la podemos resumir aquí:

		x_1	x_2	x_3	x_4
Solución x^1		0	0	2	5
$\nabla f(x^1)$		-4	-6	0	0
$\nabla_B f(x^1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	x_3	1	1	1	0
	x_4	1	5	0	1
r		-4	-6	0	0

Tabla 3.4: Tabla asociada a x^1

Ahora, de 3.15, $\alpha = \text{máx}\{-r_1, -r_2, -r_3, -r_4\} = -r_2 = 6$. Además, de 3.16, $\beta = \text{máx}\{x_3r_3, x_4r_4\} = 0$; por lo tanto, de 3.17, $\nu = 2$. Podemos ver que $-r_2 = 6$ implica que x_2 puede aumentarse para producir un valor menor de la función objetivo. La dirección de búsqueda viene dada por 3.18 y 3.20. De 3.18 tenemos $d_N = (d_1, d_2)^t = (0,1)$; y de 3.20 obtenemos $d_B^t = (d_3, d_4) = -(1,5)$. Por lo tanto, $d^1 = (0,1,-1,-5)^t$.

Movimiento en la dirección de búsqueda: A partir de $x^1 = (0,0,2,5)^t$, queremos minimizar la función objetivo en la direción $d^1 = (0,1,-1,-5)^t$. El valor máximo de λ tal que $x^1 + \lambda d^1$ sea factible, se calcula usando 3.21, y obtenemos $\lambda_{max} = \text{mín}\{\frac{2}{1}, \frac{5}{5}\} = 1$. Por tanto, $\lambda_1 = 1$ y $x^2 = x^1 + \lambda d^1 = (0, 1, 1, 0)^t$.

Iteración 2:

Dirección de búsqueda: En el punto $x^2 = (0, 1, 1, 0)^t$, por 3.12, tenemos $I_2 = \{2, 3\}$, $B = [a_2, a_3]$ y $N = [a_1, a_4]$. Además, debemos tener en cuenta que $\nabla f(x^2) = (-6, -2, 0, 0)^t$, y por 3.14:

$$r^t = (-6, -2, 0, 0) - (0, -2) \begin{bmatrix} \frac{4}{5} & 0 & 1 & -\frac{1}{5} \\ \frac{1}{5} & 1 & 0 & \frac{1}{5} \end{bmatrix} = (-\frac{28}{5}, 0, 0, \frac{2}{5})$$

	x_1	x_2	x_3	x_4
Solución x^2	0	1	1	0
$\nabla f(x^2)$	-6	-2	0	0
$\nabla_B f(x^2) = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \begin{array}{c} x_3 \\ x_2 \end{array}$	$\frac{4}{5}$ $\frac{1}{5}$	0 1	1 0	$-\frac{1}{5}$ $\frac{1}{5}$
r	$-\frac{28}{5}$	0	0	$\frac{2}{5}$

Tabla 3.5: Tabla asociada a x^2

Ahora, de 3.15 y 3.16, $\alpha = \text{máx}\{-r_1, -r_2, -r_3\} = -r_1 = \frac{28}{5}$ y $\beta = \text{máx}\{x_2 r_2, x_3 r_3, x_4 r_4\} = 0$; y por lo tanto $\nu = 1$. Esto significa que x_1 puede ser aumentado. De 3.18 y 3.20, tenemos $d_N = (d_1, d_4)^t = (1, 0)^t$ y $d_B = (d_3, d_2)^t = (\frac{4}{5}, -\frac{1}{5})^t$. Por lo tanto, $d^2 = (1, -\frac{1}{5}, -\frac{4}{5}, 0)^t$.

Movimiento en la dirección de búsqueda: A partir de $x^2 = (0, 1, 1, 0)^t$, queremos minimizar la función objetivo en la direción $d^2 = (1, -\frac{1}{5}, -\frac{4}{5}, 0)^t$. El valor máximo de λ para que $x^2 + \lambda d^2$ sea factible, se calcula usando 3.21 como sigue: $\lambda_{max} = \text{mín}\{\frac{1}{\frac{1}{5}}, \frac{5}{\frac{4}{5}}\} = \frac{5}{4}$, y por tanto, $\lambda_2 = \frac{35}{31}$ y $x^3 = x^2 + \lambda d^2 = (\frac{35}{31}, \frac{24}{31}, \frac{3}{31}, 0)^t$.

Iteración 3:

Dirección de búsqueda: Teniendo $x^3 = (\frac{35}{31}, \frac{24}{31}, \frac{3}{31}, 0)^t$, de 3.12, tenemos $I_2 = \{1, 1\}$, $B = [a_1, a_2]$ y $N = [a_3, a_4]$. Además, también tenemos que $\nabla f(x^3) = (-\frac{32}{31}, -\frac{160}{31}, 0, 0)^t$, y por 3.14:

$$r^t = (-\frac{32}{31}, -\frac{160}{31}, 0, 0) - (-\frac{32}{31}, -\frac{160}{31}) \begin{bmatrix} 1 & 1 & \frac{5}{4} & -\frac{1}{4} \\ 0 & 0 & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} = (0, 0, 0, \frac{32}{31})$$

En este caso, $\alpha = \text{máx}\{-r_1, -r_2, -r_3\} = 0$ y $\beta = \text{máx}\{x_1 r_1, x_2 r_2, x_3 r_3, x_4 r_4\} = 0$; y por lo tanto, el punto $x^3 = (\frac{35}{31}, \frac{24}{31}, \frac{3}{31}, 0)^t$ es una solución **KKT** y, en consecuencia, óptima para el problema.

		x_1	x_2	x_3	x_4
Solución x^3		$\frac{35}{31}$	$\frac{24}{31}$	$\frac{3}{31}$	0
$\nabla f(x^3)$		$-\frac{32}{31}$	$-\frac{160}{31}$	0	0
$\nabla_B f(x^3) = \begin{bmatrix} -\frac{32}{31} \\ -\frac{160}{31} \end{bmatrix}$	x_1	1	0	$\frac{5}{4}$	$-\frac{1}{4}$
	x_2	0	1	$-\frac{1}{4}$	$\frac{1}{4}$
r		0	0	0	$\frac{32}{31}$

Tabla 3.6: Tabla asociada a x^3

Podemos ver gráficamente el proceso del algoritmo en la figura 3.2.

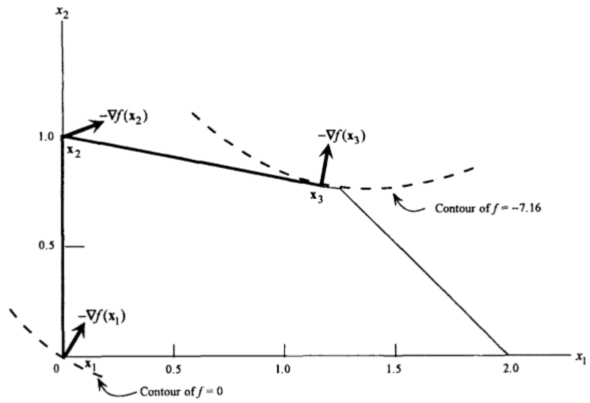


Figura 3.2: Proceso del método Simplex-Convexo

3.3.3. Convergencia

La convergencia del método Simplex-Convexo a un punto **KKT** se puede establecer de una manera muy similar a como lo hicimos en el método del Gradiente Reducido, como veremos en el siguiente teorema.

Teorema

Sea $f : R^n \to R$ una función continuamente diferenciable, y consideramos un problema de la forma 3.11.

A es una matriz $m \times n$ y $b \in R^m$ un vector tal que todos los puntos extremos de la región factible tiene m componentes positivas, y cualquier conjunto de las $m-$columnas de A son linealmente independientes. Suponemos que $\{x_k\}$ está generado por el algoritmo de Gradiente Reducido. Entonces, cualquier punto de acumulación es un punto **KKT**.

Como vimos en el método anterior, debemos observar que hay dos suposiciones cruciales: la elección de las variables independientes y la de la dirección de búsqueda. Además, este teorema no proporciona información sobre la velocidad de convergencia, que puede variar dependiendo de las características específicas del problema de optimización. Sin embargo, si la función es convexa y no lineal, el mínimo de la función se va a alcanzar y va a ser único, con lo cual el método convergerá irremediablemente hacia él.

3.4. Método de Dirección Alterna de Multiplicadores

El método de Dirección Alterna de Multiplicadores [2] (ADMM, por sus siglas en inglés) es un algoritmo utilizado para resolver problemas de optimización convexa que involucran gran cantidad de variables y restricciones. ADMM es particularmente efectivo para problemas de programación convexa debido a sus propiedades de escalabilidad. Estos problemas se pueden resolver eficientemente a medida que aumenta su tamaño. Esto es útil para el campo de aprendizaje automático, como veremos más adelante.

ADMM se basa en la descomposición del problema original en subproblemas más manejables. Cada subproblema se resuelve en paralelo con la ayuda de un multiplicador de Lagrange, que introduce una penalización por violar las restricciones del problema. El algoritmo utiliza una estrategia de iteración de dirección alterna, el cual actualiza iterativamente la solución de cada subproblema mientras se mantiene fijo el resto.

3.4.1. Resumen del método de Dirección Alterna de Multiplicadores

ADMM, para abreviar, resuelve problemas de la forma

$$\begin{aligned} \text{mín } & f(x) + g(z) \\ \text{s.a: } & Ax + Bz = c \\ & x \in X, z \in Z \end{aligned} \tag{3.22}$$

donde $A \in R^{p\times n}, B \in^{p\times m}$, $c \in R^p$, y X y Z son conjuntos convexos cerrados. Se asume que $f : R^n \to R$ y $g : R^n \to R$ son funciones convexas. El valor óptimo del problema 3.22 viene dado por

$$p^* = \text{ínf}\{f(x) + g(z)|Ax + Bz = c, x \in X, z \in Z\}$$

Teniendo ya nuestro problema definido, introducimos su función del lagrangiano aumentado:

$$L\rho(x, y, z) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \qquad (3.23)$$

donde $\rho > 0$.

El método de Dirección Alterna de Multiplicadores, dadas las iteraciones actuales $(x_k, z_k, u_k) \in R^n, R^m, R^m$, siendo u_k el multiplicador de Lagrange, genera unas nuevas iteraciones $(x_{k+1}, z_{k+1}, u_{k+1})$, minimizando primero el lagrangiano aumentado con respecto a x, luego con respecto a z, y por último realizando una actualización del multiplicador u.

$$x_{k+1} = \arg\min_{x \in R^n} L_p(x, z_k, u_k), \qquad (3.24)$$

$$z_{k+1} = \arg\min_{z \in R^n} L_p(x_{k+1}, z, u_k), \qquad (3.25)$$

$$u_{k+1} = u_k + \rho(Ax_{k+1} + Bz_{k+1} - c)), \qquad (3.26)$$

donde $\rho > 0$. Este proceso se continúa hasta que se cumple un criterio de convergencia que nosotros decidamos.

A continuación, aplicaremos este método en un ejemplo, para entender bien su estructura y funcionamiento.

3.4.2. Aplicación en Machine Learning

La Programación Convexa es un área fundamental en el aprendizaje automático(Machine Learning) y el análisis de datos. Muchos problemas de optimización en estas áreas pueden ser formulados como problemas de programación convexa, y el ADMM [6] es una técnica popular para resolverlos de manera eficiente. Debido al aumento en el tamaño y complejidad de los conjuntos de datos modernos, es cada vez más determinante poder abordar los problemas con una gran cantidad de funciones.

Ejemplo

Un ejemplo común de la aplicación del ADMM en el aprendizaje automático es en la Regresión de Lasso. Este un tipo de problema de regresión lineal en el que se intenta ajustar un modelo a un conjunto de datos, pero con la particularidad de que se busca minimizar el error de ajuste sujeto a una restricción adicional sobre los coeficientes del modelo. Esta restricción adicional consiste en que la suma de los valores absolutos de los coeficientes no supere un cierto valor máximo preestablecido.

El objetivo de esta restricción es evitar el sobreajuste del modelo y mejorar su capacidad de generalización.

Si formulamos el problema de Regresión de Lasso como un problema de optimización convexa, tenemos:

$$\text{mín } \tfrac{1}{2}\|Ax-b\|_2^2 + \tau\|z\|_1$$

donde $f(x) = \frac{1}{2}\|Ax-b\|_2^2$ y $g(x) = \tau\|z\|_1$, siendo ambas dos funciones convexas y cerradas.

Por tanto, ahora tenemos el problema de la forma 3.22:

$$\begin{aligned} &\text{mín } f(x) + g(z) \\ &\text{s.a: } x - z = 0 \end{aligned}$$

y podemos resolverlo haciendo uso del método de Dirección Alterna de Multiplicadores.

1. Primero, escribimos la función del lagrangiano aumentado correspondiente:

$$L\rho(x, y, z) = \frac{1}{2}\|Ax-b\|_2^2 + \tau\|z\|_1 + y^T(x-z) + \frac{\rho}{2}\|x-z\|_2^2$$

2. Teniendo esto, podemos dividir el problema en tres subproblemas más sencillos que se resuelven iterativamente:

$$x_{k+1} = \arg\min_{x\in R^n} L_p(x, z_k, y_k) = \arg\text{mín } \frac{1}{2}\|Ax-b\|_2^2 + y_k^T(x) + \frac{\rho}{2}\|x-z_k\|_2^2$$

$$z_{k+1} = \arg\min_{z\in R^n} L_p(x_{k+1}, z, y_k) = \arg\text{mín } \tau\|z\|_1 - y^T z + \frac{\rho}{2}\|x_{k+1}-z\|_2^2$$

$$u_{k+1} = u_k + \rho(x_{k+1-z_{k+1}})$$

3. El proceso iterativo del paso 2 se repite hasta que se alcanza la convergencia del algoritmo.

El resultado final es un vector de coeficientes x que indica la importancia relativa de cada variable en la predicción de nuestro modelo de regresión.

3.4.3. Convergencia

Una de las propiedades más importantes del método de Dirección Alterna de Multiplicadores es su capacidad para converger a una solución óptima en un número finito de pasos. La siguiente proposición da el principal resultado de convergencia, y su demostración puede encontrarse en la Sección 5.4 de [7].

Proposición

Consideramos el problema

$$\begin{aligned} &\text{mín } f(x) + g(z) \\ &\text{s.a: } Ax + Bz = c \\ &\quad\;\; x \in X, z \in Z \end{aligned} \tag{3.27}$$

donde $A \in R^{p\times n}, B \in^{p\times m}$, $c \in R^p$, X y Z son conjuntos convexos cerrados y $f : R^n \to R$ y $g : R^n \to R$ son funciones convexas. Supongamos que existe un par de soluciones óptimas primal y dual, y que, o $dom(f)$ es compacto, o $A'A$ es invertible. Entonces:

- La sucesión $\{x_k, z_k, u_k\}$ generada por 3.24, 3.25 y 3.26 está acotada, y cada punto límite de $\{x_k\}$ es una solución óptima del problema. Además, $\{u_k\}$ converge a una solución óptima dual.
- La sucesión residual $\{Ax_k - b\}$ converge a 0, y si además se cumple que $A'A$ es invertible, entonces converge a una solución óptima primal.

Esta propiedad de convergencia garantiza que, bajo ciertas condiciones, el método puede encontrar una solución óptima de manera confiable y en un número finito de pasos, como hemos visto. Es importante tener en cuenta que la elección de los puntos iniciales puede afectar la velocidad de la convergencia.

4

Ejemplos de problemas de optimización convexa

Entre los diferentes problemas de optimización convexa que podemos encontrarnos, se encuentran casos especiales como, por ejemplo, el problema de mínimos cuadrados (LS, least squares), o los problemas cuadráticos (QP, quadratic program). Vamos a condierar algunos de estos problemas y vamos a resolverlos haciendo uso de CVXPY (tomado de [8] y [9]), un paquete de modelado integrado en Python, de código abierto, para problemas de optimización convexa.

4.1. Problemas de mínimos cuadrados

Un problema de mínimos cuadrados (o regresión lineal) en optimización convexa es un tipo de problema en el que se busca encontrar la solución que minimice la distancia cuadrática entre una función dada y un conjunto de datos observados. Es decir, tenemos las medidas: $A \in R^{m\times n}$ y $b \in R^m$ y buscamos un vector $x \in R^n$ tal que Ax está próximo de b. La proximidad se define como la suma de las diferencias al cuadrado:

$$\textstyle\sum_{i=1}^{m}(a_i^T x - b_i)^2),$$

también conocido como la l_2-norma cuadrada, $\|Ax - b\|_2^2$.

4.1.1. Aplicación

Para un mayor entendimiento del problema planteado, exponemos un ejemplo real. Por ejemplo, supongamos que una empresa desea minimizar los costos de producción de sus productos utilizando un proceso de fabricación que tiene varias entradas: la cantidad de materiales y la cantidad de horas de trabajo. Se pueden recopilar datos de los costos de producción y la cantidad de materiales utilizados en varios ciclos de producción. El objetivo es encontrar la combinación óptima de materiales para minimizar los costos de producción.

Esto se puede escribir como un problema de programación convexa de la siguiente manera:

$$\text{mín } \|Ax - b\|_2^2 \tag{4.1}$$

donde tenemos las siguientes variables:

- $x \in R^n$ es un vector que representa las cantidad de cada material total.
- $A \in R^{m \times n}$ es una matriz que contiene la cantidad de materiales utilizados en cada ciclo de producción.
- $b \in R^m$ es un vector que contiene los costos de producción observados en cada ciclo.

Observamos que la función objetivo es una función convexa ya que está formada por una norma, y éstas son siempre funciones convexas por la desigualdad triangular.

Sea x^* la solución optima, el valor $r = Ax^* - b$ se conoce como residual o el error. Si $\|r\|_2 = 0$, podemos decir que tenemos un ajuste perfecto.

Procedemos a resolver ejemplo planteado 4.1 con el paquete de python CVXPY.

```
# Import packages.
import cvxpy as cp
import numpy as np
```

Rellenamos los datos de A y b de manera aleatoria.

```
m = 3 #3 materiales diferentes
n = 4 #4 ciclos
A = np.array([[5, 7, 6, 8],[6, 8, 7, 9],[ 7, 9, 8, 10]])
b = np.array([500, 400, 600])
```

Ahora, resolvemos el problema con CVXPY:

```
x = cp.Variable(n)
cost = cp.sum_squares(A @ x - b)
constraints = [x>=0]
prob = cp.Problem(cp.Minimize(cost), constraints)
prob.solve()

# Print result.
print("\nThe optimal value is", prob.value)
print("The optimal x is")
print(x.value)
print("The norm of the residual is ", cp.norm(A @ x - b, p=2).value)
```

Que nos da como resultado:

```
 The optimal value is 15061.224489795917
The optimal x is
[7.48411940e-20 2.58071074e-20 4.38229744e-22 5.55102041e+01]
The norm of the residual is  122.72418054236874
```

El vector x es la cantidad de cada material utilizado para minimizar el costo total, que se corresponde al valor óptimo de 15061.22.

4.2. Problemas de programación cuadrática

Un programa cuadrático es un problema de optimización con un objetivo cuadrático y restricciones afines de igualdad y desigualdad. Una forma estándar común es la siguiente:

$$\begin{array}{l} \text{mín } \frac{1}{2}x^T P x + q^T x \\ \text{s.a: } Ax \leq b \end{array} \tag{4.2}$$

Aquí, $P \in S^{+n}$, $q \in R^{+n}$, $q \in R^n$,$A \in R^{m\times n}$ y $b \in R^m$, son datos del problema, y $x \in R^n$ el vector a optimizar.

La función objetivo es cuadrática y convexa, ya que la matriz P es simétrica y definida positiva. Las funciones que definen las restricciones forman un conjunto convexo.

Cuando resolvemos un problema cuadrático, además de una solución x^*, obtenemos una solución dual λ^* correspondiente a las restricciones de desigualdad. Una entrada positiva de λ_i^* indica que la restricción $a_i^T x \leq b_i$ se mantiene con la igualdad para x^* y sugiere que cambiando b_i podría cambiar el valor óptimo.

4.2.1. Aplicación

Un ejemplo real de un problema cuadrático puede ser la asignación óptima de recursos limitados en una empresa que produce varios productos utilizando diferentes máquinas. Supongamos que una empresa tiene tres máquinas y produce dos tipos de productos: A y B. Cada producto tiene un costo de producción y un beneficio de venta diferentes, y cada máquina tiene una capacidad máxima de producción y un costo de uso asociado.

El objetivo de la empresa es minimizar el costo total de producción, sujeto a las restricciones de capacidad y costos de uso de las máquinas. En nuestro problema, los datos serían:

- x es un vector de tamaño 6 que representa la cantidad de cada producto producido en cada máquina.

- P es una matriz simétrica y definida positiva que representa los costos de producción y beneficios de venta de cada producto.
- q es un vector de tamaño 6 que representa los costos de uso de cada máquina.
- A es una matriz de tamaño 3x6 que representa la capacidad máxima de producción de cada máquina.
- b es un vector de tamaño 3 que representa la cantidad total de producción requerida de cada producto.

Al resolver este problema de programación convexa, la empresa puede determinar la cantidad óptima de cada producto que debe producirse en cada máquina minimizando el costo total de producción, mientras se cumplen las restricciones de capacidad y costos de uso de las máquinas.

Resolvemos este problema de la forma 4.2 con ayuda del paquete de Python CVXPY.

Primero generamos los datos de manera aleatoria:

```
# Import packages.
import cvxpy as cp
import numpy as np
# Generate a random non-trivial quadratic program.
m = 3
n = 6
p = 3
np.random.seed(9)
P = np.random.randn(n, n)
P = P.T @ P
q = np.random.randn(n)
A = np.random.randn(p, n)
b = A @ np.random.randn(n)
```

Ahora, resolvemos el problema con CVXPY:

```
x = cp.Variable(n)
prob = cp.Problem(cp.Minimize((1/2)*cp.quad_form(x, P) + q.T @ x),
                 [A @ x <= b,
                  x >= 0])
prob.solve()

# Print result.
print("\nThe optimal value is", prob.value)
print("A solution x is")
print(x.value)
```

Que nos da como resultado:

```
 The optimal value is 2.052711801349038
A solution x is
[ 1.19873021e+00 3.92837010e-23 2.68191163e-23  3.44887215e-22
  3.56633428e-01  7.72536995e-01]
A dual solution corresponding to the inequality constraints is
[0.         1.08165948 0.07747114]
```

donde el vector x representa la cantidad de cada producto producido en cada máquina, y el valor óptimo 2.05 el costo total de producción.

4.3. Problemas de programación semidefinida

Un problema de programación semidefinida (SDP) es un problema de optimización convexa de la forma

$$\begin{aligned} &\min\, tr(CX) \\ &\text{s.a: } tr(A_i X) = b_i, i = 1, \ldots, p \\ &\quad X \geq 0 \end{aligned} \tag{4.3}$$

donde **tr** es la función *traza* (es decir, la suma de los elementos diagonales de una matriz), $X \in S^n$ es la matriz a optimizar, $C, A_1, \ldots, A_p \in S^n$, y $b_1, \ldots, b_p \in R$ son datos del problema. Además, S^n denota el conjunto de matrices simétricas $n \times n$.

4.3.1. Aplicación

Un ejemplo real de optimización convexa de un problema de programación semidefinida es la optimización del diseño de un filtro de señal. En este problema, se busca diseñar un filtro que elimine ciertas frecuencias no deseadas de una señal mientras se preservan las frecuencias deseadas.

Tenemos las siguientes variables:

- X es la matriz de covarianza de la señal filtrada
- C es una matriz de pesos que penaliza las frecuencias no deseadas
- A_i son matrices de observación que se utilizan para estimar la energía de las frecuencias en la señal original
- b_i son los valores objetivo de la energía de las frecuencias en la señal filtrada.

La solución del problema de programación semidefinida proporciona los valores óptimos de la matriz de covarianza del filtro, que se pueden utilizar para diseñar un filtro que cumpla con los requisitos de la señal. La restricción $X \geq 0$ garantiza que la matriz de covarianza resultante sea semidefinida positiva, lo que es necesario para asegurar que la señal filtrada tenga un espectro de potencia realista.

Resolvemos el ejemplo planteado de la forma 4.3 con ayuda de el paquete CVXPY.

Primero, importamos los paquetes correspondientes y generamos los datos de manera aleatoria:

```
import cvxpy as cp
import numpy as np
n = 3
p = 3
np.random.seed(1)
C = np.random.randn(n, n)
A = []
b = []
for i in range(p):
    A.append(np.random.randn(n, n))
    b.append(np.random.randn())
X = cp.Variable((n,n), symmetric=True)
```

Ahora, resolvemos el problema con CVXPY:

```
constraints = [X >> 0] # >> denotes matrix inequality.
constraints += [
    cp.trace(A[i] @ X) == b[i] for i in range(p)
]
prob = cp.Problem(cp.Minimize(cp.trace(C @ X)),
                  constraints)
prob.solve()

# Print result.
print("The optimal value is", prob.value)
print("A solution X is")
print(X.value)
```

Que nos da como resultado:

```
The optimal value is 2.654348003008652
A solution X is
[[ 1.6080571  -0.59770202 -0.69575904]
 [-0.59770202  0.22228637  0.24689205]
 [-0.69575904  0.24689205  1.39679396]]
```

donde la matriz X es la matriz de covarianza óptima de la señal filtrada que cumple los requisitos del problema, con su respectivo valor óptimo.

4.4. Problemas de control

La Optimización Convexa se puede utilizar para resolver muchos problemas que surgen en los procesos de control.

Por ejemplo, supongamos que queremos diseñar un controlador óptimo para un sistema de navegación de vehículos autónomos. El estado del sistema se puede representar como un vector que incluye la posición, la velocidad y la orientación del vehículo. El controlador toma una entrada que representa la aceleración y el ángulo de dirección del vehículo. El objetivo del controlador es llevar el vehículo a un destino específico en el menor tiempo posible, sujeto a restricciones en la velocidad máxima, la aceleración máxima, el radio de giro mínimo, y otros factores.

Este problema de control se puede formular como un problema de optimización de la forma

$$\begin{aligned} &\min \textstyle\sum_{t=0}^{T-1} \ell(x_t, u_t) + \ell_T(x_T) \\ &\text{s.a: } x_{t+1} = Ax_t + Bu_t \\ &\qquad (x_t, u_t) \in \mathcal{C}, \quad x_T \in \mathcal{C}_T, \end{aligned} \tag{4.4}$$

donde $A \in \mathbf{R}^{n\times n}$ y $B \in \mathbf{R}^{n\times m}$ son matrices conocidas, $\ell(x_t, u_t)$ y $\ell_T(x_T)$ funciones convexas y $(x_t, u_t) \in \mathcal{C}$ y $x_T \in \mathcal{C}_T$ son conjuntos convexos.

Podemos asumir que nos encontramos frente a un problema de optimización convexa ya que:

- $x_{t+1} = Ax_t + Bu_t$ es un sistema lineal, y por tanto, convexo.
- La función objetivo es una suma de funciones convexas.

4.4.1. Aplicación

Vamos a tratar de resolver el ejemplo planteado. Los datos son:

- x_t representa el estado del vehículo en el momento t, y u_t la entrada de control en el momento t.
- $A \in \mathbf{R}^{n\times n}$ es la matriz de dinámica que describe cómo evoluciona el sistema en el tiempo.
- $B \in \mathbf{R}^{n\times m}$ es la matriz de entrada que describe cómo la entrada de control afecta al sistema.
- C es el conjunto de restricciones en el estado y entrada de control, y C_T es el conjunto de restricciones en el estado final.
- $\ell : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}$ es el coste de cada etapa, y ℓ_T es el coste final.

El conjunto de restricciones C_T puede ser diseñado para garantizar que el vehículo llegue al destino en un estado seguro. La solución óptima proporciona el controlador óptimo que minimiza la función de costo sujeto a las restricciones dadas.

Aplicamos el paquete CVXPY para resolver este problema.

En el siguiente código resolvemos un problema de control con $n = 8$, $m = 2$ y $T = 50$. Las matrices A y B y el estado inicial x_0 se eligen aleatoriamente (con $A \approx I$). Utilizamos el costo de etapa $\ell(x, u) = \|x\|_2^2 + \|u\|_2^2$, la restricción de entrada $\|u_t\|\infty \leq 1$, y la restricción $x_T = 0$.

Primero, generamos los datos:

```
import numpy as np
import cvxpy as cp
np.random.seed(1)
n = 8
m = 2
T = 50
alpha = 0.2
beta = 3
A = np.eye(n) - alpha * np.random.rand(n, n)
B = np.random.randn(n, m)
x_0 = beta * np.random.randn(n)
```

Ahora, resolvemos el problema con CVXPY:

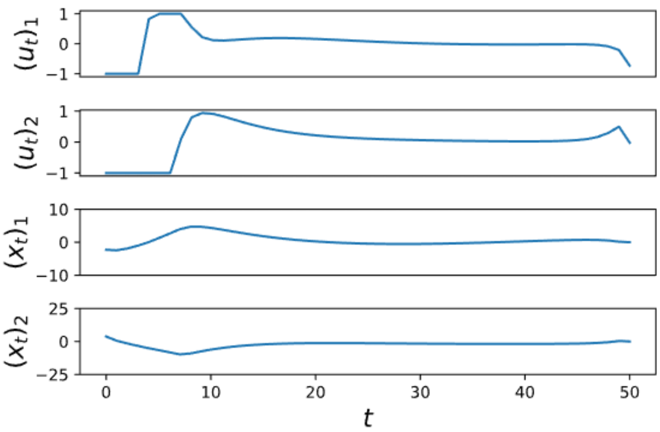
```
x = cp.Variable((n, T + 1))
u = cp.Variable((m, T))

cost = 0
constr = []
for t in range(T):
    cost += cp.sum_squares(x[:, t + 1]) + cp.sum_squares(u[:, t])
    constr += [x[:, t + 1] == A @ x[:, t] + B @ u[:, t],
               cp.norm(u[:, t], "inf") <= 1]
constr += [x[:, T] == 0, x[:, 0] == x_0]
problem = cp.Problem(cp.Minimize(cost), constr)
problem.solve()
```

Que nos da como resultado:

```
2515.656065442021
```

En el siguiente gráfico, mostramos u_1, u_2, x_1 y x_2 en función del tiempo t en las diferentes iteraciones. Observamos que u_t da un salto importante, lo que muestra que la restricción de entrada es significativa.


$(u_t)_1$
1
0
−1
$(u_t)_2$
1
0
−1
$(x_t)_1$
10
0
−10
$(x_t)_2$
25
0
−25
0
10
20
30
40
50
t

Conclusiones

Los problemas de optimización convexa son de gran importancia en muchas áreas. Esto se debe a que los estos problemas tienen propiedades matemáticas muy deseables, como la unicidad de la solución óptima y la facilidad de encontrarla. Hemos visto las diferentes condiciones de optimalidad y hemos presentado las condiciones necesarias y suficientes de optimalidad para la optimización convexa, incluyendo la condición de Karush-Kuhn-Tucker para la programación no lineal. Estas condiciones son cruciales para garantizar la optimalidad de las soluciones y permiten identificar soluciones factibles óptimas.

Además, se han presentado varios métodos de resolución de problemas de programación convexa, como el método del Subgradiente Reducido, el método de Simplex Convexo de Zangwill y el método de Dirección Alterna de Multiplicadores. Cada uno de estos métodos tiene sus propias ventajas y desventajas, por lo que es importante elegir el más adecuado para el problema específico que se está tratando.

Por último, se han resuelto varios ejemplos de problemas de optimización convexa, como el problema de mínimos cuadrados (LS, least squares), la programación semidefinida, los problemas de control y la programación cuadrática.

En definitiva, la Programación Convexa es un área de la Optimización muy importante y útil en una amplia gama de aplicaciones. A través de esta memoria, hemos tratado de comprender un poco mejor los conceptos de convexidad y la importancia de la Optimización Convexa en la resolución de problemas complejos de optimización.

Bibliografía

[▶]

[1] BAZARAA, Mokhtar. *Nonlinear Programming, Theory and Algorithms.* 3rd edition. John Wiley & Sons, Inc., Hoboken, New Jersey. 2006.

[2] BERTSEKAS, Dimitri P. *Convex Optimization Algorithms.* Athena Scientific, Belmont, Massachusetts. 2015.

[3] BOYD, Stephen. VANDENBERGHE, Lieven . *Convex Optimization.* Cambridge University Press. ISMN 0 521 83378 7 2004.

[4] NESTEROV, Yurii. *Lectures on Convex Optimization.* 2nd Edition. Springer. 2018.

[5] BRINKHUIS, Jan. *Convex Analysis for Optimization. A Unified Approach.* Springer. 2020.

[6] BOYD, Stephen. PARIKH, Neal. CHU, Eric. PELEATO, Borja. ECKSTEIN, Jonathan. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.* Now Foundations and Trends. 2011.

[7] BERTSEKAS, Dimitri P. TSITSIKLIS, John N. *Parallel and Distributed Computation: Numerical Methods* republished by Athena Scientific, Belmont, MA, 1997. On line at http://web.mit.edu/dimitrib/www/pdc.html.

[8] DIAMOND, Steven. BOYD Stephen. CHU, Eric. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research.* 2016, vol 17, n 17, pp. 1–5. Disponible en: https://web.stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf.

[9] AGRAWAL, Akshay and VERSCHUEREN, Robin and DIAMOND, Steven and BOYD, Stephen. A rewriting system for convex optimization problems *Journal of Control and Decision.* 2018, vol 15, pp. 42–60. Disponible en: https://web.stanford.edu/~boyd/papers/pdf/cvxpy_rewriting.pdf

[10] Package Python *Python.* https://www.cvxpy.org/.

Convex Programming Problems

Julia Alicia Palenzuela Rodríguez
Facultad de Ciencias • Sección de Matemáticas
Universidad de La Laguna
alu0101027722@ull.edu.es

Abstract

The aim of this dissertation is to provide a complete and detailed review of the theory and methods for solving convex optimization problems, as well as their application in different situations. To achieve this objective, the dissertation begins by defining basic concepts of the theory of convex sets and functions. Then, it delves into convex optimization problems and their optimality conditions. Various resolution methods are explained with illustrative examples. Finally, the application of the concepts and methods studied in different situations is exemplified, with the aim of providing the reader with a comprehensive and useful understanding of the topic.

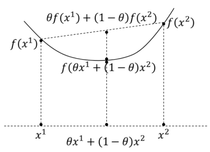
1. Introduction

Convex optimization is a branch of mathematical optimization that focuses on finding the minimum of a convex function subject to certain constraints. A subset is convex if and only if it contains the line segments connecting any pair of points. In other words:

$$\forall x_1, x_2 \in S, \forall \lambda \in [0,1], x_1 + \lambda(x_2 - x_1) \in S$$

A function is convex if the straight line connecting any pair of points on the function lies above the function at all points between those two points, i.e.

$$\forall x_1, x_2 \in S, \forall \lambda \in [0,1], x_1 + \lambda(x_2 - x_1) \in S$$



2. Convex optimization problems

A convex optimization problem has the form:
Let $f : S \to R$ and:

$$\begin{aligned} \min \ & f(x) \\ \text{s.a: } & x \in S \end{aligned} \tag{1}$$

where $f(x)$ is a convex function and $S \subseteq R^n$ is a convex set.
Optimal conditions
One of the most important optimality conditions in convex optimization is that a local minimum will always be a global minimum. This provides us with a great deal of advantages.
The Karush-Kuhn-Tucker (KKT) conditions are a set of necessary conditions for constrained optimization. These conditions extend the Lagrange optimality conditions to nonlinear and nonsmooth optimization problems.
In summary, the KKT conditions state that for a point to be optimal, certain conditions must be satisfied:

- The constraints of the problem must be satisfied at the optimal point.
- The gradient of the objective function evaluated at the optimal point must be proportional to the weighted sum of the gradients of the constraints evaluated at the same point.
- The slack variables (variables introduced to transform nonlinear constraints into linear constraints) must be non-negative.
- The product of the slack variables and the corresponding constraints must be equal to zero.

Together, these conditions ensure that the optimal point satisfies both the problem constraints and the optimality conditions of the objective function.

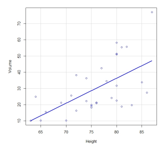
3. Methods of solving convex programming problems and examples

We explore three methods for solving convex programming problems. The methods we examine are:

1. Reduced Subgradient Method
2. Zangwill's Convex-Simplex Method
3. The Alternating Direction Method of Multipliers

Each of these methods approaches convex programming problems from a different perspective and offers unique advantages and disadvantages in terms of convergence, execution speed, and algorithmic complexity. Through practical examples and detailed explanations, we explore how these methods are applied to solve convex programming problems and evaluate their effectiveness in different scenarios.
We also see some examples of different convex programming problems.

min $\|Ax - b\|_2^2$.



$$\begin{aligned} \min \ & \tfrac{1}{2} x^T P x + q^T x \\ \text{s.a: } & Ax \leq b \end{aligned}$$

$$\begin{aligned} \min \ & \textstyle\sum_{t=0}^{T-1} \ell(x_t, u_t) + \ell_T(x_T) \\ \text{s.a: } & x_{t+1} = Ax_t + Bu_t \\ & (x_t, u_t) \in \mathcal{C}, \quad x_T \in \mathcal{C}_T, \end{aligned}$$

References

[1] BAZARAA, Mokhtar. *Nonlinear Programming, Theory and Algorithms.* 3rd edition. 2006.

[2] BERTSEKAS, Dimitri P. *Convex Optimization Algorithms.* Athena Scientific, Belmont, Massachusetts. 2015.

[3] BOYD, Stephen. VANDENBERGHE, Lieven *Convex Optimization.* Cambridge University Press. 2004.

Sección de Matemáticas
Universidad de La Laguna