UNIVERSIDAD MIGUEL
HERNÁNDEZ DE ELCHE

FACULTAD DE CIENCIAS SOCIALES Y JURÍDICAS

TRABAJO DE FIN DE GRADO

Constante de Hoffman

Autora
Teresa Juan Bru

Tutora
María Josefa Cánovas
Cánovas

GRADO DE ESTADÍSTICA EMPRESARIAL

Curso 2023/2024

Resumen

El presente Trabajo de Fin de Grado se centra en el estudio de la constante de Hoffman.

Con el fin de proporcionar un contexto adecuado, se comienza explicando que son los sistemas de desigualdades lineales, poniendo especial énfasis en el concepto de consistencia. La consistencia es crucial, ya que la constante de Hoffman, sobre la que versa este trabajo, proporciona una cota superior sobre las tasas de variación de soluciones factibles con respecto a variaciones de los parámetros del sistema (en nuestro caso, los términos independientes del mismo), suponiendo que este es consistente. Además se ha desarrollado un código en MATLAB que analiza la consistencia de dichos sistemas.

A continuación, se abordará que es la constante de Hoffman y como se calcula. Se presenta un código en MATLAB diseñado para calcular esta constante de manera eficiente.

Finalmente, se estudia una de las múltiples aplicaciones de la constante de Hoffman en el ámbito de la programación lineal, demostrando así su utilidad en aplicaciones reales.

Índice

1 Introducción

El presente Trabajo de Fin de Grado versa sobre la constante de Hoffman.

Con el fin de entender el sentido de esta constante y de comprender las causas que motivaron su actual repercusión, se incluyen breves notas históricas.

La constante de Hoffman, que lleva el nombre del matemático Alan J. Hoffman, es un concepto crucial en el campo de la programación lineal. Hoffman, nacido el 30 de mayo de 1924 en Nueva York y que falleció el 18 de enero de 2021, contribuyó notablemente a la geometría, optimización combinatoria, teoría de grafos, entre otros, y recibió numerosos premios y distinciones a lo largo de su carrera: En particular, fue reconocido "IBM felow" en 1977, miembro de la Academia Nacional de Ciencias de EEUU en 1982 y de la Academia Americana de Artes y Ciencias en 1987 y recibió el premio John von Neumann en 1992. Además, fue editor fundador de la revista científica Linear Algebra and its Applications.

Hoffman también trabajó en programación lineal y en la teoría de la estabilidad de sistemas de desigualdades lineales, donde se enmarca el actual Trabajo de Fin de Grado. En el contexto de la programación lineal, Hoffman es conocido, entre otras cosas, por haber desarrollado el primer ejemplo en el que se produce un ciclo al aplicar el método simplex, lo que marcó un hito importante en su evolución y relevancia en la optimización.

En un breve documento técnico del National Bureau of Standards (NBS, ahora el National Institute of Standards and Technology, en Washington DC), Hoffman establece que un punto no factible que "casi" satisface el conjunto de desigualdades lineales de un sistema determinado está "cerca" de algún otro punto que sí es factible, bajo ciertas nociones de "casi" y "cerca" que se formalizarán a lo largo del trabajo. En términos informales, la constante de Hoffman proporciona una cota superior del cociente entre la variación de puntos factibles y la variación de los términos independientes del sistema (entendidos como parámetros del modelo). Dicha constante tiene implicaciones importantes en el diseño y la implementación de algoritmos de programación lineal (para detalles, véase el trabajo de Peña, Vera y Zuluaga [9]. Numerosos investigadores han contribuido al desarrollo de fórmulas que permitieran determinar esta constante, de entre ellos citamos Klatte y Thiere [7], Peña, Vera y Zuluaga [9] (véase Camacho, Cánovas y Parra [4] para su extensión a sistemas con una cantidad infinita de desigualdades lineales).

En el segundo capítulo estudiaremos los sistemas de desigualdades lineales, centrándonos especialmente en la consistencia de estos, ya que se trata de un aspecto fundamental para poder calcular la constante de Hoffman. Analizaremos el modelo paramétrico asociado a estos sistemas y presentaremos un teorema que establece las condiciones necesarias y suficientes para que un sistema de desigualdades lineales sea consistente.

Además se ha elaborado en MATLAB un código que nos permite llevar a cabo un análisis de la consistencia. Presentaremos un modelo de optimización cuadrática que nos permitirá determinar la consistencia de un sistema de desigualdades lineales mediante la función quadprog de MATLAB.

Llevar a cabo este análisis de la consistencia es importante para interpretar la constande de Hoffman, ya que la consistencia de sistemas está implicita en la definición de dicha constante.

En el tercer capítulo se aborda qué es la constante de Hoffman y cómo se calcula. Destacamos el hecho de que este trabajo presenta una fórmula para calcular dicha constante que está basada exclusivamente en los datos del sistema, concretamente, en los coeficientes del miembro izquierdo del sistema, que se consideran fijos (los términos independientes son considerados como parámetros, sujetos a posibles perturbaciones). Este hecho nos permite diseñar una implementación para calcularla. De hecho, una de la principales aportaciones del trabajo actual consiste en la elaboración de un código en MATLAB que nos proporciona el valor exacto de dicha constante.

El último capítulo se centra en estudiar una aplicación particular de la constante de Hoffman dentro del ámbito de la programación lineal. Comenzamos analizando cómo acotar la tasa de variación del valor óptimo cuando cambiamos los parámetros del término independiente en un problema de programación lineal resoluble. El teorema principal de esta sección establece una cota superior para la variación del valor óptimo bajo perturbaciones del término independiente. Esta cota está determinada por un término que incluye la norma del vector de coeficientes de la función objetivo y la constante de Hoffman.

Para ilustrar este concepto, se presenta un ejemplo práctico de un problema específico de programación lineal, el problema de la dieta. Este problema nos ayuda a demostrar la utilidad del teorema en aplicaciones reales.

Finalmente comentaremos que este trabajo se ha elaborado tras haber cursado asignaturas del Grado de Estadística Empresarial que proporcionan la base necesaria de programación lineal y cuadrática, por lo que se presupone un conocimiento básico de estas materias. Existen en la literatura numerosos textos sobre estas disciplinas de entre los que citamos Bazaraa, Jarvis y Sherali [2], Bertsimas [3], Barbolla, Cerdá y Sanz [1], Goberna, Jornet y Puente [5] y Luenberger [8].

2 Sistemas de desigualdades lineales

2.1 Introducción

En este capítulo nos adentraremos en el estudio de los sistemas de desigualdades lineales, enfocándonos especialmente en el concepto de consistencia, un aspecto crucial para interpretar la constante de Hoffman que se abordará en el próximo capítulo.

Comenzaremos describiendo el modelo paramétrico asociado a estos sistemas, donde cada desigualdad está parametrizada con respecto a los términos independientes. Analizaremos la estructura de estos sistemas, así como la multifunción conjunto factible asociada.

Para comprender la consistencia de estos sistemas, presentaremos un teorema que establece las condiciones necesarias y suficientes para que un sistema de desigualdades lineales sea consistente. También ilustraremos estas condiciones con un ejemplo.

Por último, nos sumergiremos en la implementación del análisis de la consistencia utilizando MATLAB. Presentaremos un modelo de optimización cuadrática que nos permitirá determinar la consistencia de un sistema de desigualdades lineales, y detallaremos la sintaxis y los pasos necesarios para llevar a cabo este análisis utilizando la función quadprog de MATLAB.

Este análisis de la consistencia es importante para interpretar la constante de Hoffman, ya que esta propiedad está implícita en la definición de dicha constante. El estudio de la constante de Hoffman constituye el objetivo del capítulo 3. Así pues, este capítulo desarrolla las herramientas teóricas y computacionales necesarias para el desarrollo de los siguientes capítulos.

2.2 Descripción del modelo paramétrico

A lo largo de este trabajo, consideramos sistemas de desigualdades lineales parametrizados con respecto al miembro de la derecha. Formalmente consideramos:

$$\sigma(b) = \{a_i'x \leq b_i, i = 1, \ldots, m\}, \tag{1}$$

con $m \in \mathbb{N}$ y donde $x \in \mathbb{R}^n$ es el vector de variables, $b = (b_i)_{i=1}^m \in \mathbb{R}^m$ es el parámetro (vector de términos independientes) del sistema y $a_i \in \mathbb{R}^n$, $i = 1, \ldots, m$. Por defecto, los vectores de $\mathbb{R}^k$ con $k \in \mathbb{N}$ se consideran vectores columna, y la prima representa la traspuesta. Con el fin de escribir el sistema de forma matricial, introducimos la matriz A de tamaño $m \times n$ cuya i-ésima fila es a_i'.

Asociado al sistema parametrizado (1) consideramos la llamada función conjunto factible $\mathcal{F} : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ (donde "$\rightrightarrows$ " indica que se trata de una multifunción, es decir, una aplicación que asocia a cada $b \in \mathbb{R}^m$ un subconjunto de $\mathbb{R}^n$ llamado conjunto factible) definido como:

$$\mathcal{F}(b) = \{x \in \mathbb{R}^n | Ax \leq b\}. \tag{2}$$

En lo que sigue, 0_k representa el vector nulo de $\mathbb{R}^k$, $k \in \mathbb{N}$.

Ejemplo:

En $\mathbb{R}^2$, consideramos A= $\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}$. La función conjunto factible asociada será:

$$\mathcal{F}(b) = \{x \in \mathbb{R}^2 | x_1 + x_2 \leq b_1, x_1 + 2x_2 \leq b_2, 2x_1 + x_2 \leq b_3, \}$$

donde b= $\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, x= $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

Seguidamente presentamos una ilustración gráfica del conjunto factible (en color gris) para el valor del parámetro $b = 0_3$:

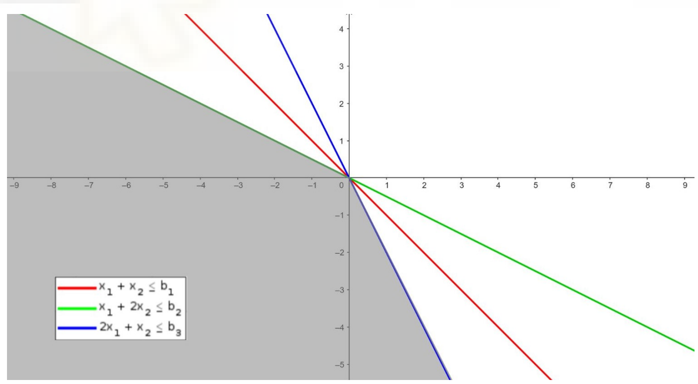

Figure 1: Elaboración propia

Respecto de la topología, en el espacio de parámetros, $\mathbb{R}^m$, consideramos la norma del supremo, $\|b\|_\infty = \max\{|b_i|, i = 1, \ldots, m\}$.

En el espacio de las variables, $\mathbb{R}^n$, consideramos la norma euclídea que está dada por:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

donde $x = (x_1, x_2, \ldots, x_n)' \in \mathbb{R}^n$.

2.3 Caracterización de la consistencia

Se dice que $\sigma(b)$ es consistente si $\mathcal{F}(b) \neq \emptyset$. Seguidamente, introducimos un resultado de caracterización de la consistencia:

Teorema: Sea $b \in \mathbb{R}^m$. Las siguientes condiciones son equivalentes:

i) $\sigma(b)$ es consistente;

ii)$\nexists \lambda_1, \lambda_2, \ldots, \lambda_m \geq 0$ tal que:

$$\begin{pmatrix} 0_n \\ -1 \end{pmatrix} = \sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix}.$$

Ejemplo:
Consideremos el sistema asocioado a los elementos $b = (-1, -1. -1)'$ y $A = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$.

Veamos que es inconsistente, esto es, que existen $\lambda_1, \lambda_2, \lambda_3 \geq 0$ tal que:

$$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = \lambda_1 \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} + \lambda_2 \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} + \lambda_3 \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}$$

.

Resolvemos el sistema:

$$\left.\begin{array}{rcl} 0 & = & \lambda_1 - \lambda_2 \\ 0 & = & \lambda_1 - \lambda_3 \\ -1 & = & -\lambda_1 - \lambda_2 - \lambda_3 \end{array}\right\} \quad \begin{array}{r} \lambda_1 = \lambda_2 \\ \lambda_2 = \lambda_3 \\ -1 = -3\lambda_3 \end{array}$$

En consecuencia, $\lambda_3 = \frac{1}{3}$ y, por lo tanto, $\lambda_1 = \lambda_2 = \frac{1}{3}$.

A continuación con el fin de implementar en MATLAB la comprobación de la consistencia, plantearemos un modelo de optimización cuya solución determina si el sistema $\sigma(b)$ es consistente o no.

Concretamente, consideramos el problema:

$$(P_0)\ Min \quad \left\| \begin{pmatrix} 0_n \\ -1 \end{pmatrix} - \sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix} \right\|_2^2$$

$$s.a. \quad \lambda_i \geq 0 \quad i = 1, \ldots, m.$$

Nota: Nótese que se cumple la condición (ii) del teorema anterior si y solo si el valor óptimo de (P_0) es positivo. En otras palabras, $\sigma(b)$ es consistente si el valor óptimo de (P_0) es positivo.

Con el fin de especificar la sintaxis de MATLAB, veamos la expresión matricial del modelo (P_0). En primer lugar, nótese que para $z \in \mathbb{R}^k, k \in \mathbb{N}$, se tiene:

$$\|z\|_2^2 = \sum_{i=1}^{k} z_i^2 = z'z = \begin{pmatrix} z_1 & \cdots & z_k \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix}.$$

Por otro lado, podemos escribir:

$$\sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix} = \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda, \quad \text{donde } \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix}.$$

En consecuencia,

$$\left\| \begin{pmatrix} 0_n \\ -1 \end{pmatrix} - \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda \right\|_2^2 = \left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix} - \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda \right)' \left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix} - \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda \right)$$

$$= \left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix}' - \lambda' \begin{pmatrix} A & b \end{pmatrix} \right) \left(\begin{pmatrix} 0_n \\ -1 \end{pmatrix} - \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda \right)$$

$$= 1 - \begin{pmatrix} 0_n \\ -1 \end{pmatrix}' \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda - \lambda' (A\ b) \begin{pmatrix} 0_n \\ -1 \end{pmatrix} + \lambda' (A\ b) \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda$$

$$= 1 - b'\lambda - \lambda' b + \lambda'(A\ b) \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda$$

$$= 1 - 2b'\lambda + \lambda'(A\ b) \begin{pmatrix} A' \\ b' \end{pmatrix} \lambda.$$

Se trata de un problema de programación cuadrática que resolveremos con la función quadprog. Seguidamente, por completitud, introducimos la notación necesaria sobre programación cuadrática y su sintaxis en MATLAB.

2.4 Implementación del análisis de la consistencia en MATLAB

La notación comúnmente empleada para describir problemas de programación cuadrática es la siguiente:

$$\begin{array}{lll} (P) & Min & \frac{1}{2} x'Qx + c'x \\ & s.a & a_i' x \leq b_i,\ i = 1, ..., m, \end{array}$$

donde:

- Q es una matriz cuadrada simétrica,
- $c, a_i \in \mathbb{R}^n$, $i = 1, ..., m$,
- $b = (b_1, ..., b_m)' \in \mathbb{R}^m$,
- $x \in \mathbb{R}^n$ es el vector de variables del problema.

La notación empleada en lo que sigue se ajusta a la de la ayuda de MATLAB (nótese que H se corresponde con Q; f con c y, además, se consideran sistemas de desigualdades y ecuaciones).

La función quadprog admite diferentes argumentos de entrada y salida. La versión más sencilla es "`>> x=quadprog(H,f,A,B)`", donde `x` es la solución propuesta por MATLAB del problema:

$$\begin{array}{ll} Min & \frac{1}{2}\texttt{x'Hx+f'x} \\ s.a & \texttt{Ax} \leq \texttt{B}. \end{array}$$

Puede ampliarse el número de argumentos de entrada en los formatos:

- "`>> x=quadprog(H,f,A,B,Aeq,Beq)`", que resuelve el problema anterior añadiendo como restricciones el sistema de ecuaciones '`Aeq x = Beq`',
- "`>> x=quadprog(H,f,A,B,Aeq,Beq,LB,UB)`" añade al anterior el vector de cotas inferiores `LB` y de cotas superiores `UB`; esto es, `LB <= x <= UB.`

Asimismo, puede ampliarse el número de argumentos de salida. En nuestro caso nos interesa el formato:

- "`>> [x,v]=quadprog(H,f,A,B,Aeq,Beq,LB,UB)`

puesto que el valor de v indica si el sistema e consistente o no. Concretamente, dado que la función de MATLAB obvia las constantes en la función objetivo, resolveremos el siguiente modelo sin constante:

$$\begin{array}{ll} Min & \frac{1}{2}\lambda\texttt{'H}\lambda\texttt{+f'}\lambda \\ s.a & \lambda \geq \texttt{0}, \end{array}$$

donde:

$$H = 2(Ab)\begin{pmatrix} A' \\ b' \end{pmatrix} \ y \ f = -2b.$$

Para ello, empleamos la sintaxis:

- "`>> [lambda,v]=quadprog(H,f,[ ],[ ],[ ],[ ],zeros(1,m),[ ]).`

En el apéndice A se encuentra el código que solicita al usuario los datos del sistema, A y b, e indica como salida si el sistema es consistente o no. Nótese que, con la notación actual, el sistema será consistente si v+1>0.

Ejemplo:

Para verificar la consistencia de los siguientes sistemas de ecuaciones lineales, hemos utilizado el código previamente mencionado.

Para $b = (-1, -1. -1)'$, $A = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$, obtenemos que el sistema es inconsistente.

Debemos introducir las matrices y llamar a la función:

```
% Definir matrices A, B y vector b
A = [1 1; -1 0; 0 -1 ];
b = [-1; -1;-1];

% Llamar a la función resolverQuadprog
[x, v] = resolverQuadprog(A, b);
```

Figure 2: Código de entrada

MATLAB nos devolverá el siguiente resultado:

```
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>
El sistema es inconsistente.
```

Figure 3: Salida de MATLAB

Para $b = (0,0,0,0)'$, $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & -\frac{1}{2} \end{pmatrix}$, obtenemos que el sistema es consistente.

Introducimos las matrices y llamamos a la función.

```
% Definir matrices A, B y vector b
A=[1,0;0,1;1,1;-1,-1/2];
b=[0;0;0;0]

% Llamar a la función resolverQuadprog
[x,v] = resolverQuadprog(A, b);
```

Figure 4: Código de entrada

El resultado obtenido es el siguiente.

```
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>
El sistema es consistente.
```

Figure 5: Salida de MATLAB

3 Constante de Hoffman

3.1 Introducción

En este capítulo, se aborda qué es la constante de Hoffman y cómo se calcula.

Para empezar, la sección 3.2 presenta el Lema de Hoffman, que formaliza la idea expresada en la introducción de este trabajo de que un punto no factible para un sistema dado que "casi" verifica las restricciones de dicho sistema, está "cerca" del conjunto factible del mismo, es decir, existe una constante k, que acota la distancia entre dicho punto y el conjunto factible.

La constante de Hoffman se define como el mínimo de k que cumple el Lema de Hoffman cuando el sistema es consistente. A continuación se detalla el procedimiento para calcular esta constante, denotada como $\mathrm{Hof}(A)$.

Por último, se presenta el código desarrollado en MATLAB para calcular la constante de Hoffman y se aplica a varios ejemplos para ilustrar su uso práctico.

3.2 Lema de Hoffman

El siguiente teorema recoge el celebrado Lema de Hoffman, que puede encontrarse en [6], anunciado en la introducción de este capítulo. En lo que sigue $[a]_+$ representa la parte positiva del número real a, esto es, $[a]_+ = max\{a, 0]$.

Teorema: Dados $a_1, \ldots, a_m \in \mathbb{R}^n$ existe una constante $k \geq 0$ tal que:

$$d(z, \mathcal{F}(b)) \leq k \max\{[a_i' z - b_i]_+,\ i = 1, \ldots, m\} \tag{3}$$

para todo $z \in \mathbb{R}^n$ y todo $b \in \mathbb{R}^m$ tal que $\sigma(b) = \{a_i' z \leq b_i, i = 1, \ldots, m\}$ es consistente.

Def: Dados $a_1, \ldots, a_m \in \mathbb{R}^n$, llamamos constante de Hoffman al infimo (de hecho, mínimo) de las constantes $k \geq 0$ tal que se cumple (3) para cualquier $z \in \mathbb{R}^n$ y cualquier $b \in \mathbb{R}^m$ tal que $\sigma(b)$ es consistente.

El siguiente teorema proporciona una caracterización de la constante de Hoffman, apelando a perturbaciones del miembro derecho de las restricciones, en vez de a la máxima violación de las mismas.

Teorema: Dados $a_1, \ldots, a_m \in \mathbb{R}^n$, la constante de Hoffman coincide con el infimo (de hecho, mínimo) de las constantes $k \geq 0$ tales que para cualesquiera $b^1, b^2 \in \mathbb{R}^m$, verificando que $\sigma(b^1)$ y $\sigma(b^2)$ son consistentes, se tiene:

$$d(x^1, \mathcal{F}(b^2)) \leq k \left\| b_1 - b_2 \right\|_\infty .$$

Nótese que la constante de Hoffman solo depende de $A = \begin{pmatrix} a_1' \\ \vdots \\ a_m' \end{pmatrix}$ y, en consecuencia, se denota por $\mathrm{Hof}(A)$.

A continuación, presentamos un resultado que proporciona una forma operativa de calcular la constante de Hoffman, en tanto que involucra exclusivamente a ciertos datos (coeficientes del miembro derecho) del sistema.

Teorema: Dados $a_1, \ldots, a_m \in \mathbb{R}^n$, se tiene que:

$$\mathrm{Hof}(A) = \max_{\substack{J \subset \{1,\ldots,m\} \\ 0_n \notin \mathrm{conv}\{a_i,\ i \in J\}}} d\left(0_n, \mathrm{conv}\left\{a_i,\ i \in J\right\}\right)^{-1} .$$

Ejemplo:

Calculamos $\mathrm{Hof}(A)$ para la siguiente familia de sistemas parametrizada respecto del miembro derecho de las restricciones:

$$\begin{aligned} x_1 &\leq b_1, \\ x_2 &\leq b_2, \\ x_1 + x_2 &\leq b_3, \\ -x_1 - \frac{1}{2}x_2 &\leq b_4. \end{aligned}$$

Nótese que podemos restringirnos a aquellos subconjuntos $J \subsetneq \{1, \ldots, m\}$ tales que $0_n \notin conv\{a_i, i \in J\}$ y J es maximal con respecto al orden establecido por contenido de conjuntos (no existe ningún $\tilde{J} \supset J$ tal que $0_n \notin conv\{a_i, i \in \tilde{J}\}$).

En este ejemplo, los conjuntos maximales de indices J tales que $\{a_i, i \in J\}$ que no incluyen el punto 0_2 en su envoltura convexa son $J = \{1, 2, 3\}$, $J = \{2, 3, 4\}$ y $J = \{1, 4\}$. En el gráfico podemos intuir que el conjunto que minimiza la distancia con el 0_2 es el $J = \{2, 3, 4\}$.

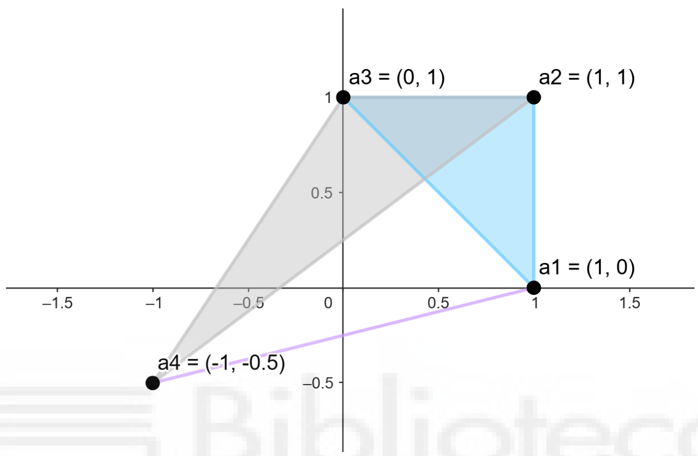

Figure 6: Elaboración propia

Sin embargo, debemos establecer un sistema para calcular Hof(A), sin necesidad de observar el gráfico. Para lograr este objetivo, hemos desarrollado un código en MATLAB, el cual se presenta a continuación.

3.3 Cálculo de la constante de Hoffman en MATLAB

El código realiza las siguientes operaciones: al introducir la matriz A, calcula todos los subconjuntos $J \subset \{1, \ldots, m\}$ que cumplen la condición $0_n \notin \operatorname{conv}\{a_i,\ i \in J\}$. Para ello, planteamos el siguiente modelo de programación cuadrática cuyo objetivo es calcular la distancia $d(0_n, conv\{a_i, i \in J\})$.

$$(P)\ Min \quad \left\| 0_n - \sum_{i \in J} \lambda_i a_i \right\|_2^2, \ sujeto\ a\ \sum_{i \in J} \lambda_i = 1, \lambda_i \geq 0.$$

Concretamente, el valor óptimo de este problema (P) proporciona $d(0_n, conv\{a_i, i \in J\})^2$. Así pues, si dicho valor es positivo se confirma que $0_n \notin conv\{a_i, i \in J\}$. Además, en una segunda etapa, calculando el inverso del mínimo de estas distancias, obtenemos la constante de Hoffman. Seguidamente, detallamos las matrices que emplearemos como datos en la función quadprog y para resolver (P).

La matriz A_J está formada por las filas de la matriz A cuyas posiciones pertenecen al subconjunto J, cuya sintaxis en MATLAB es $A_J = A(J,:)$. En la función quadprog introducimos la siguiente matriz Q, $Q = 2\ A'_J A_J$.

Este proceso permite identificar la constante de Hoffman y su subconjunto asociado J.

Ejemplo:

Calculamos la constante de Hoffman del ejemplo previo:

$$x_1 \leq b_1,$$

$$x_2 \leq b_2,$$

$$x_1 + x_2 \leq b_3,$$

$$-x_1 - \frac{1}{2}x_2 \leq b_4.$$

Hemos introducido en MATLAB la matriz $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & -\frac{1}{2} \end{pmatrix}$ de la siguiente manera:

```
Introduce la matriz A
[1,0;0,1;1,1;-1,-1/2]
```

Y nos devuelve lo siguiente:

```
J=

     2     3     4

Constante de Hoffman:

    0.0400
```

Como resultado, la constante de Hoffman es igual a 0.04 y el subconjunto J correspondiente es $\{2, 3, 4\}$.

Ejemplo:

Calculamos Hof(A) para el siguiente sistema parametrizado:

$$
\begin{aligned}
x_1 &\leq b_1, \\
x_2 &\leq b_2, \\
x_3 &\leq b_3, \\
x_4 &\leq b_4, \\
x_1 + x_2 + x_3 + x_4 &\leq b_5, \\
-x_1 - x_2 - x_3 - x_4 &\leq b_6, \\
x_1 + x_2 &\leq b_7, \\
-x_1 - 0.5x_2 - 0.5x_3 - 0.2x_4 &\leq b_8.
\end{aligned}
$$

Introducimos en MATLAB la matriz A.

$$
A = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 \\
1 & 1 & 0 & 0 \\
-1 & -0.5 & -0.5 & -0.2
\end{pmatrix}
$$


```
Introduce la matriz A
[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1;1,1,1,1;-1,-1,-1,-1;1,1,0,0;-1,-0.5,-0.5,-0.2]
```

Y nos devuelve el siguiente resultado:

```
J=
     1     2     3     6     7     8
Constante de Hoffman:
    0.0044
```

La constante de Hoffman es igual a 0.0044 y el subconjunto J asociado es $\{1, 2, 3, 6, 7, 8\}$.

4 Aplicación a la tasa de variación del valor óptimo

4.1 Introducción

El objetivo de este capítulo es estudiar una de las múltiples aplicaciones de la constante de Hoffman en el ámbito de la programación lineal.

Para ello, primero estudiaremos cómo acotar la tasa de variación del valor óptimo. Consideramos un problema de programación lineal resoluble y nos preguntamos: ¿Cómo varía el valor óptimo cuando cambiamos los parámetros del término independiente?

El teorema principal de esta sección proporciona una cota superior para la variación del valor óptimo bajo perturbaciones de los términos independientes del sistema de restricciones. Esta variación está acotada por un término que incluye la norma del vector cuyos parámetros forman la función objetivo y la constante de Hoffman.

Para ilustrar este concepto, se presenta un ejemplo práctico de un problema específico de programación lineal, el problema de la dieta. Este problema tiene como objetivo minimizar el coste del cereal, cumpliendo con ciertos requisitos nutricionales mínimos. Se calcula la cota superior de la tasa de variación del coste óptimo del cereal cuando se modifican los parámetros del problema, demostrando así la utilidad del teorema en aplicaciones reales.

4.2 Acotación de la tasa de variación del valor óptimo

En esta sección consideramos el problema de programación lineal parametrizado con respecto al miembro de la derecha de las restricciones que formalmente escribimos como sigue:

$$\begin{array}{lll} P(b): & \text{minimizar} & c'x \\ & \text{sujeto a} & a_i'x \leq b_i, \ \ i = 1, 2, ..., m, \end{array} \tag{4}$$

donde como el secciones anteriores $x \in \mathbb{R}^n$ es el vector de variables de decisión, los vectores $c, a_i \in \mathbb{R}^n$, $i = 1, ..., m$ están fijos y $b = (b_i)_{i=1,\ldots,m} \in \mathbb{R}^m$ se considera como parámetro. Para cada $b \in \mathbb{R}^m$ consideramos el conjunto factible que, manteniendo la notación de secciones anteriores (en particular, A representa la matriz cuya $\acute{\imath}$-ésima fila es a_i'), está dado por

$$\mathcal{F}(b) := \{x \in \mathbb{R}^n \mid Ax \leq b\}, \tag{5}$$

y el valor óptimo asociado a $P(b)$ que se define como sigue:

$$v(b) := \inf\{c'x \mid x \in \mathcal{F}(b)\}. \tag{6}$$

Se dice que el problema $P(b)$ es acotado si su valor óptimo, $v(b)$, es finito. Es sabido que en programación lineal el hecho de que el problema sea acotado es equivalente a resoluble. Así pues, dado $\overline{b} \in \mathbb{R}^m$, si $-\infty < v\left(\overline{b}\right) < +\infty$ existe algún punto factible $\overline{x} \in \mathcal{F}\left(\overline{b}\right)$ tal que $c'\overline{x} = v\left(\overline{b}\right)$.

El siguiente teorema proporciona una cota superior sobre la variación del valor óptimo bajo perturbaciones del parámetro b. En la prueba, se hace uso de la conocida desigualdad de Cauchy-Schwarz, que en $\mathbb{R}^n$ con norma euclídea se escribe como

$$|u'v| \leq \|u\|_2 \|v\|_2, \text{ para } u, v \in \mathbb{R}^n.$$

Teorema: Consideremos el problema parametrizado (4), y sea $\mathrm{Hof}(A)$ la constante de Hoffman asociada a la función conjunto factible (5). Para cualesquiera b_1, $b_2 \in \mathbb{R}^m$ tales que $P(b_1)$ y $P(b_2)$ son problemas resolubles se tiene que

$$|v(b_1) - v(b_2)| \leq \|c\|_2 \operatorname{Hof}(A) \|b_1 - b_2\|_\infty. \tag{7}$$

Demostración: Sean b^1, $b^2 \in \mathbb{R}^m$ tales que $P\left(b^1\right)$ y $P\left(b^2\right)$ son resolubles. Sean $x^1 \in$ y x^2 soluciones óptimas de $P\left(b^1\right)$ y $P\left(b^2\right)$, respectivamente; esto es, $x^1 \in \mathcal{F}\left(b^1\right)$ y $x^2 \in \mathcal{F}\left(b^2\right)$, y verifican $c'x^1 = v\left(b^1\right)$ y $c'x^2 = v\left(b^2\right)$. Entonces, podemos escribir

$$v(b_1) - v(b_2) = v(b_1) - c'x^2.$$

Por otro lado, sea $\widehat{x}^1 \in \mathcal{F}\left(b^1\right)$ la mejor aproximación de x^2 en $\mathcal{F}\left(b^1\right)$, lo que significa que

$$d\left(x^2, \mathcal{F}\left(b^1\right)\right) = \left\|\widehat{x}^1 - x^2\right\|_2.$$

Como $v\left(b^1\right)$ es el valor óptimo de $P\left(b^1\right)$, y $\widehat{x}^1$ es un punto factible del mismo problema, se tiene que

$$v(b_1) \leq c'\widehat{x}^1.$$

En consecuencia, se tiene

$$\begin{array}{rcl} v(b_1) - v(b_2) & \leq & c'\widehat{x}^1 - c'x^2 = c'\left(\widehat{x}^1 - x^2\right) \leq \|c\|_2 \left\|\widehat{x}^1 - x^2\right\|_2 \\ & = & \|c\|_2 \, d\left(x^2, \mathcal{F}\left(b^1\right)\right) \leq \|c\|_2 \operatorname{Hof}(A) \|b_1 - b_2\|_\infty. \end{array}$$

Intercambiando los papeles que b_1 y b_2 se tiene con un razonamiento análogo

$$v(b_2) - v(b_1) \leq \|c\|_2 H(A) \|b_1 - b_2\|_\infty.$$

Por lo que se concluye

$$|v(b_1) - v(b_2)| \leq \|c\|_2 \operatorname{Hof}(A) \|b_1 - b_2\|_\infty.$$

Ejemplo:

Para comprender mejor esta sección, presentaremos un ejemplo que ilustra el resultado anterior.

$$\begin{array}{ll} \text{Min} & -x_1 - x_2 \\ \text{s.a.} & \quad x_1 \leq b_1, \\ & \quad x_2 \leq b_2, \\ & \quad x_1 + x_2 \leq b_3, \\ & -x_1 - \frac{1}{2}x_2 \leq b_4. \end{array}$$

Para resolver este problema, consideramos la matriz A dada por:

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & -\frac{1}{2} \end{pmatrix}$$

.

Aplicamos el teorema (7) :

$$\left|v\left(b^1\right) - v\left(b^2\right)\right| \leq \|c\|_2 \operatorname{Hof}(A) \|b_1 - b_2\|_\infty . \tag{8}$$

Sabemos que $c = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$, por tanto, $\|c\|_2 = \sqrt{2}$ y que la constante de Hoffman Hof(A)=0.04, cuyo valor hemos calculado en la sección 3.3 con el código de MATLAB.

Para $b^1, b^2 \in \mathbb{R}^4$ cualesquiera:

$$\left|v\left(b^1\right) - v\left(b^2\right)\right| \leq \sqrt{2}\ 0.04 \|b_1 - b_2\|_\infty .$$

La expresión $\|c\|_2 \operatorname{Hof}(A)$ nos proporciona una cota superior de la tasa de variación del valor óptimo con respecto a variaciones de los términos independientes del sistema.

4.3 Aplicación al problema de la dieta

El teorema que establece (7) estudiado en la sección anterior puede aplicarse, entre otros contextos, al problema de la dieta.

El problema de la dieta representa una de las primeras aplicaciones de la programación lineal, y comenzó a utilizarse en los hospitales para determinar la dieta más económica con la que alimentar a los pacientes a partir de unas especificaciones nutritivas mínimas. En la actualidad también se aplica con éxito en el ámbito agrícola con la misma idea de encontrar la combinación óptima de alimentos que, logrando un aporte nutritivo mínimo, suponga el menor coste posible. A continuación, utilizaremos un ejemplo concreto de este tipo de problema.

Problema de selección de grano:

Un centro de nutrición utiliza tres tipos de granos para elaborar un cereal natural que vende por kilos. El coste y el contenido de proteínas, hidratos de carbono, fósforo y magnesio por kg. de cada tipo de grano se muestran en la siguiente tabla:

GRANO	COSTE POR KG.	PROTEINAS (unidades/kg)	HIDRATOS C (unidades/kg.)	FÓSFORO (unidades/kg)	MAGNESIO (unidades/kg)
A	0.33 €	22	16	8	5
B	0.47 €	28	14	7	0
C	0.38 €	21	25	9	6

Los requisitos nutricionales mínimos por día para un adulto son 3 unidades de proteínas, 2 de hidratos de carbono, 1 de fósforo, y 0.425 de magnesio. Se tratará pues de establecer la mezcla adecuada de granos que logra cubrir estas necesidades con el mínimo coste para el centro. Supondremos que se ha de emplear como mínimo 1 unidad de cada tipo de grano.

Planteamiento del problema

Seguidamente definimos las variables del modelo, restricciones y objetivo.

- Las variables de decisión :
  - x_1 : cantidad de grano tipo A (en kg.)
  - x_2 : cantidad de grano tipo B (en kg.)
  - x_3 : cantidad de grano tipo C (en kg.)

- A continuación presentamos las restricciones que determinan los requisitos nutricionales mínimos, teniendo en cuenta que ha de emplearse una unidad como mínimo en cada tipo de grano:
  - Requisito nutricional de proteína: $22x_1 + 28x_2 + 21x_3 \geq 3$,
  - Requisito nutricional de hidratos de carbono: $16x_1+14x_2+25x_3 \geq 2$,
  - Requisito nutricional de fósforo: $8x_1 + 7x_2 + 9x_3 \geq 1$,
  - Requisito nutricional de magnesio: $5x_1 + 6x_3 \geq 0.425$,
  - Unidades mínimas : $x_1, x_2, x_3 \geq 1$.

Finalmente, el objetivo es minimizar el coste, que para las cantidades x_1, x_2 y x_3 de los granos A, B y C, respectivamente, viene dado por $0.33x_1 + 0.47x_2 + 0.38x_3$.

Con todo, el **modelo de programación lineal** lo podemos escribir como sigue:

$$\begin{array}{ll} \text{Minimizar} & 0.33x_1 + 0.47x_2 + 0.38x_3 \\ \text{sujeto a} & 22x_1 + 28x_2 + 21x_3 \geq 3, \\ & 16x_1 + 14x_2 + 25x_3 \geq 2, \\ & 8x_1 + 7x_2 + 9x_3 \geq 1, \\ & 5x_1 + 6x_3 \geq 0.425, \\ & x_1, x_2, x_3 \geq 1. \end{array}$$

Queremos calcular la cota superior de la tasa de variación del valor óptimo con respecto a variaciones de los términos independientes. En otras palabras, deseamos determinar cuanto puede variar el coste si se modifican los términos independientes.

Para ello, aplicamos el teorema (7):

$$|v(b_1) - v(b_2)| \leq \|c\|_2 \operatorname{Hof}(A) \|b_1 - b_2\|_\infty .$$

Calculamos la norma de $c = \begin{pmatrix} 0.33 \\ 0.47 \\ 0.38 \end{pmatrix}$:

$$\|c\|_2 = \sqrt{0.33^2 + 0.47^2 + 0.38^2} = \sqrt{0.474} = 0.688$$

Para calcular Hof(A), introducimos la matriz A, en el código del Apéndice B,

$$A = \begin{pmatrix} -22 & -28 & -21 \\ -16 & -14 & -25 \\ -8 & -7 & -9 \\ -5 & 0 & -6 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

de la siguiente manera:

```
Introduce la matriz A
[-22,-28,-21;-16,-14,-25;-8,-7,-9;-5,0,-6;-1,0,0;0,-1,0;0,0,-1]
```

y nos devuelve Hof$(A) = 0.3333$:

```
J=

     1     2     3     4     5     6     7

Constante de Hoffman:

    0.3333
```


Por tanto, obtenemos:

$$|v(b_1) - v(b_2)| \leq 0.688 \times 0.3333 \, \|b_1 - b_2\|_\infty \, , \; |v(b_1) - v(b_2)| \leq 0.229 \, \|b_1 - b_2\|_\infty$$

Así 0.299 es la cota superior de la tasa de variación del coste óptimo del grano si varían los términos independientes del problema.

5 Apéndice A

Análisis de la consistencia

```
function [x, v, consistente] = resolverQuadprog(A, b)
    % Tama o de A para determinar el n mero de
       variables
    [m, ~] = size(A);
    AB_concatenado = [A, b];
    matriz_concatenada = vertcat(A', b');

    % Definir las matrices H, A_eq y b_eq seg n las
       especificaciones dadas
    H = 2 * AB_concatenado * matriz_concatenada;
    c = 2 * b;
    lb = zeros(m, 1);

    % Resolvemos el sistema
    [x, v] = quadprog(H, c, [], [], [], [], lb);

    % Determinar si el sistema es consistente
    if (v > -1)
        disp('El sistema es consistente.');
    else
        disp('El sistema es inconsistente.');
    end
end
```

6 Apéndice B

Constante de Hoffman

```
A = input('Introduce la matriz A');

% Obtener el n mero de filas de A (que es m)
m = size(A, 1);

% Definir el vector J que indica las filas de A (de 1
   a m)
J = 1:m;

% Combinaciones procesadas
combinaciones_procesadas = [zeros(1,m)];
valores = [];

% Encontrar las combinaciones m s grandes de J que no
    contienen el punto (0,0) en su envolvente convexa
for k = m:-1:1
    combinaciones = nchoosek(J, k); % Generar todas
       las combinaciones de tama o k de los n meros
       de fila en J

    for j = 1:size(combinaciones, 1)
        combinacion_actual = combinaciones(j, :);
        if size(combinacion_actual, 2) < m
            combinacion_actual = [combinacion_actual,
               zeros(1, m - size(combinacion_actual,
               2))];
        end
        seg = 1 ;
        for i = 1:size(combinaciones_procesadas,1)
            int = intersect(combinacion_actual,
               combinaciones_procesadas(i,:));
            int2 = int(find(int));
            if size(int2, 2) < m
                int2 = [int2, zeros(1, m - size(int2,
                   2))];
            end

            if combinacion_actual == int2
                seg = 0;
            end
```

```
        end

        if seg == 1
            % Extraer las filas correspondientes a la
                combinaci n actual
            AJ = A(combinaciones(j, :), :)';
            H = 2 * (AJ' * AJ);
            f = zeros(size(AJ, 2), 1);
            Aeq = ones(size(AJ, 2), 1)';
            beq = ones(size(Aeq, 1), 1);
            lb = zeros(size(AJ, 2), 1);
            [x, v] = quadprog(H, f, [], [], Aeq, beq,
               lb, []);

            if v > 0.000001
                combinaciones_procesadas = [
                   combinaciones_procesadas;
                   combinacion_actual];
                combinaciones_procesadas;
                valores = [valores; v];
            end

        end
    end
end

combinaciones_procesadas = combinaciones_procesadas(2:
   size(combinaciones_procesadas, 1), :);
valores;

% Encontrar el valor m nimo y la combinaci n
   correspondiente
[min_valor, idx] = min(valores);
combinacion_minima = combinaciones_procesadas(idx, :);

% Encuentra el  ltimo   ndice  no cero
sin0 = find(combinacion_minima, 1, 'last');

% Recorta el vector hasta el  ltimo   ndice  no cero
J = combinacion_minima(1:sin0);

% Mostrar los resultados
disp('J=');
disp(J);
disp('Constante de Hoffman:');
disp(min_valor);
```

7 Bibliografía

References

[1] R. Barbolla, E. Cerdá Tena, and P. S. Á. Sanz, *Optimizacion: cuestiones, ejercicios y aplicaciones a la economia*, Prentice Hall, 2000.

[2] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Programacion lineal y flujo en redes* (2a ed.), Limusa, 1998.

[3] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Dynamic Ideas and Athena Scientific, Belmont, Massachusetts, 2008.

[4] J. Camacho, M. J. Cánovas, and J. Parra, *From calmness to Hoffman constants for linear inequality systems*, SIAM J. Optim., vol. 32, pp. 2859–2878, 2022.

[5] M. A. Goberna Torrent, V. Jornet Plá, and R. Ó. Puente, *Optimizacion lineal: teoria, metodos y modelos*, McGraw Hill, 2004.

[6] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Research Nat. Bur. Standards, vol. 49, pp. 263–265, 1952.

[7] D. Klatte and G. Thiere, *Error bounds for solutions of linear equations and inequalities*, Z. Oper. Res., vol. 41, pp. 191–214, 1995.

[8] D. G. Luenberger, *Programacion lineal y no lineal*, Addison-Wesley Iberoamericana, 1989.

[9] J. Peña, J. C. Vera, and L. F. Zuluaga, *New characterizations of Hoffman constants for systems of linear constraints*, Math. Program., vol. 187, pp. 79–109, 2021.