

Punto Interior, Predictor Corrector y Programación Cuadrática

Pablo Barranco

Diego Villegas

José Miguel Carmona

ITAM - Marzo 2020

1. Introducción

Consideremos el problema

$$\begin{aligned} &\text{Minimizar } (1/2)x^T Q x + c^T x \\ &\text{sujeto a } Ax = b \\ &\quad Fx \geq d \end{aligned}$$

donde $Q \in \mathbb{R}^{n \times n}$ es simétrica y positiva definida, $A \in \mathbb{R}^{m \times n}$ tal que $m \leq n$ y $\text{rango}(A) = m$, $F \in \mathbb{R}^{p \times n}$, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $d \in \mathbb{R}^p$. Sea $\Omega = \{x \in \mathbb{R}^n \mid Ax = b, Fx - d \geq 0\}$ el conjunto factible. Suponemos que $\Omega^0 = \{x \in \mathbb{R}^n \mid Ax = b, Fx - d > 0\} \neq \emptyset$.

Las condiciones necesarias de primer orden para un mínimo local son,

$$\mathfrak{F}(x, \lambda, \mu, z) = \begin{bmatrix} Qx + A^T \lambda - F^T \mu + c \\ Ax - b \\ -Fx + z + d \\ UZe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, (\mu, z) \geq 0 \quad (1)$$

Las condiciones perturbadas de primer orden con parámetro $\gamma > 0$ son

$$\mathfrak{F}(x, \lambda, \mu, z) = \begin{bmatrix} Qx + A^T \lambda - F^T \mu + c \\ Ax - b \\ -Fx + z + d \\ UZe - \gamma e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, (\mu, z) \geq 0 \quad (2)$$

donde $U = \text{diag}(\mu)$, $Z = \text{diag}(z)$ y $e = (1, 1, \dots, 1)^T \in \mathbb{R}^P$

1.1. Método Predictor-Corrector

Sea $w = (x, \lambda, \mu, z)$

1. Sean $w_0 = (x_0, \lambda_0, \mu_0, z_0)$, $\gamma_0 > 0$, $\gamma \approx 0$ tales que $\mu_0, z_0 > 0$. Hacemos $k \leftarrow 0$,
 $\gamma_{\text{size}} \leftarrow \frac{(\mu_0)^T(z_0)}{p}$

2. Mientras $\|F_0(w_k)\| > \text{tol}$ hacemos

- Resolvemos el sistema lineal:

$$F'_0(w_k)\Delta w_k = -F_0(w_k)$$

donde el vector solución es $\Delta w_k = (\Delta x_k, \Delta \lambda_k, \Delta \mu_k, \Delta z_k)$ es el paso de predicción.

- Determinamos $a_k \in (0, 1]$ tal que

$$z^* = z_k + \alpha_k \Delta z_k > 0, \mu^* = \mu_k + \alpha_k \Delta \mu_k > 0$$

- Hacemos:

$$\gamma \leftarrow (z^*)^T(\mu^*)/p$$

$$\gamma \leftarrow (\gamma/\gamma_{\text{size}})^3$$

$$\gamma \leftarrow \gamma * \gamma_{\text{size}}$$

- Resolvemos el sistema lineal

$$F'_0(w_k)(\Delta w_k)^* = -F_0(w_k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\Delta Z_k \Delta U_k e + \gamma e \end{bmatrix}$$

El vector $(\Delta w_k)^*$ es el paso de corrección

- Determinamos $\alpha_k \in (0, 1]$ tal que

$$z_k + \alpha_k (\Delta z_k)^* > 0, \mu_k + \alpha_k (\Delta \mu_k)^2 > 0$$

- Actualizamos

$$w_{k+1} = w_k + \alpha_k (\Delta w_k)^*$$

- $k \leftarrow k + 1$ e ir al paso 2

2. El Sistema Lineal

En el paso predictor, Δw_k y en el paso corrector, $(\Delta w_k)^*$, se debe resolver un sistema lineal de la forma:

$$\begin{bmatrix} Q & A^T & -F^T & 0 \\ A & 0 & 0 & 0 \\ -F & 0 & 0 & I_p \\ 0 & 0 & Z & U \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta \mu \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_x \\ r_\lambda \\ r_\mu \\ r_z \end{bmatrix}$$

cuya matriz es rala (tiene muchas entradas igual a cero).

Ahora mostramos que dicho sistema lineal puede reducirse a un sistema lineal de orden $(n + m)$ con la estructura:

$$\begin{bmatrix} \hat{Q} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \hat{r}_x \\ r_\lambda \end{bmatrix}$$

Realizando la multiplicación del primer sistema, obtenemos que:

$$Q\Delta x + A^T\Delta\lambda - F^T\Delta\mu = r_x \quad (3)$$

$$A\Delta x = r_\lambda \quad (4)$$

$$-F\Delta x + I_p\Delta\mu = r_\mu \quad (5)$$

$$\Delta x + Z\Delta\mu + U\Delta z = r_z \quad (6)$$

Luego, por (5):

$$\Delta\mu = r_\mu + F\Delta x$$

De (6) obtenemos que

$$\Delta\mu = Z^{-1}(r_z - U(r_\mu + F\Delta x))$$

Desarrollando la primera ecuación con lo que encontramos en (6) encontramos que

$$(Q + F^T Z^{-1} U F) \Delta x + A^T \Delta \lambda = r_x + F^T Z^{-1} r_z - F^T Z^{-1} U r_\mu$$

Siendo $\hat{Q} = Q + F^T Z^{-1} U F$, $\hat{r}_x = r_x + F^T Z^{-1} (r_z - U r_\mu)$, $Z^{-1} = \text{diag}(\frac{1}{z})$, $U = \text{diag}(U)$, obtenemos el sistema

$$\begin{bmatrix} \hat{Q} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \hat{r}_x \\ r_\lambda \end{bmatrix} \quad (7)$$

Ahora observemos el siguiente problema cuadrático:

$$\begin{aligned} & \text{Minimizar } (1/2)\Delta x^T \hat{Q} \Delta x - \hat{r}_x^T \Delta x \\ & \text{sueto a } A\Delta x = r_\lambda \end{aligned}$$

De donde los pasos predictor y corrector son calculados resolviendo un sistema lineal de la forma (7).

Su lagrangeano es el siguiente si consideramos $\Delta\lambda$ el multiplicador de lagrange.

$$\mathcal{L} = \frac{1}{2}\Delta x^T \hat{Q} \Delta x - \hat{r}_x^T \Delta x + \Delta\lambda(A\Delta x - r_\lambda)$$

Así las condiciones de primer orden son:

$$\begin{aligned} \nabla_x \mathcal{L} = 0 & \Rightarrow \hat{Q} \Delta x + A^T \Delta\lambda = \hat{r}_x \\ \nabla_{\Delta\lambda} \mathcal{L} = 0 & \Rightarrow A\Delta x = r_\lambda \end{aligned}$$

Por lo tanto, el sistema

$$\begin{bmatrix} \hat{Q} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} \hat{r}_x \\ r_\lambda \end{bmatrix}$$

son las condiciones de primer orden del problema cuadrático anterior.

3. Programación

Programamos cuatro versiones distintas que comparten la siguiente estructura:

`function[x, λ, μ, z, iter] = qpintpointpc(Q, Q, F, c, d)`

El cual resuelve el problema cuadrático por el método de puntos interiores con el esquema de preedicción y corrección bajo las siguientes condiciones:

- El número máximo de iteraciones son 100.
- La variable de salida, *iter*, es el número de iteraciones que usa el método
- Se inicia *iter* = 0 y se incrementa en uno en cada iteración
- Se detiene cuando $\|F_0(w_k)\|_2 \leq 10^{-5}$ o $k = \text{maxiter}$
- El punto inicial: $x \in \mathbb{R}^n$ satisface que $Ax = b$
- $\lambda = 0$, $\mu = z = (1, \dots, 1)^T \in \mathbb{R}^P$

3.1. Version Preliminar

Explicaremos brevemente el código realizado en clase. Primero se establecen las dimensiones de c, b, d , el maxiter, la tolerancia y la inicialización de iter en 0. Primero se resuelve el sistema lineal $Ax = b$ para obtener convergencia. Se inicializa las variables U , Z y el multiplicador de lagrange como el vector de unos de dimensión P y $\gamma = 1$. Luego, se calculan las condiciones de KunTucker.

Ahora resolvemos el sistema lineal completo. Empezamos con el lado izquierdo. Primero concatenamos Q con A^T y $-F^T$ y una matriz de ceros dimensión n, p . El segundo bloque consiste de A , una matriz cuadrada de ceros dimensión m y dos matrices de zeros dimensión m, p . El siguiente bloque es $-F$, zeros(p, m), zeros(p) y la identidad dimensión p . Por último, tenemos zeros(p, n), zeros(p, m), diag(z) y diag(u).

El lado derecho serán las condiciones de primer orden con la perturbación de la cuarta pues es el único que no constituye un sistema lineal. Resolvemos dicho sistema lineal y ahora procedemos a recortar el paso en U y en Z .

Dado $a, \Delta a \in \mathbb{R}^S$, $a > 0, \Delta a \neq 0$ determinamos $\alpha > 0$ tal que $a + \alpha \Delta a \geq 0$. Ahora, $a + \alpha \Delta a > 0 (a + \alpha \Delta a)_i \geq 0, i = 1, 2, \dots, S$ o bien puede ser que $a_i + \alpha \Delta a_i = 0 = \frac{-a_i}{\Delta a_i} \geq 0$. En este segundo caso, α siempre llega a 0 (tanto alargando o acortando). Resumiendo:

1. $\Delta a_i \geq 0 \implies$ cualquier $\alpha \geq 0$
2. $\Delta a_i < 0$ Encontramos una frontera.

Así construyo v_a tal que

$$V_{a_i} = \begin{cases} 1 & \text{si } \Delta a_i \geq 0, \\ \frac{-a_i}{\Delta a_i} & \text{si } \Delta a_i < 0, \end{cases}$$

El *alpha* que sirve para todos los componentes será

$$a = \min(V_{a_i})$$

De esta forma obtenemos las alphas que cumplan para todas las componentes como el mínimo de v_u y de v_z . Ahora creo una alpha que funcione para ambas:

$$alpha = (0,95)\min(\alpha_u, \alpha_z, 1)$$

Este 1 es una forma de precaución pues si todas las entradas resultan ser el segundo caso y mayores a 1, en realidad no estaríamos acortando el paso sino aumentándolo. Además, de forma conservadora, dicha α la multiplicamos por 0.95.

Por último podemos actualizar y obtener el nuevo punto (x, λ, u, z) . Volvemos a las condiciones de primer orden y lo evaluamos en dicho punto e iteramos gamma como $\frac{1}{2} \frac{U^T Z}{P}$ y terminamos graficando la convergencia mostrando $\|F_0(w_k)\|$ de cada iteración.

3.2. Sistema Reducido

Del sistema proveniente del metodo de Newton se obtiene un sistema reducido, dedimension ($m + n$), por lo cual es menos costoso computacionalmente. En cada iteración se chequea si la matriz reducida tiene inversa numéricamente. En caso en que no sea así se usa entonces la matriz completa.

3.3. Sistema Completo Predictor-Corrector

Para esta opción lo que hicimos fue usar el sistema completo y en la cuarta entrada del lado derecho, es decir en r_μ , no se realiza la perturbación de γ , si no que usamos predictor-corrector para hacer el recorte.

3.4. Sistema Reducido Condicional Predictor-Corrector

En este caso se combinan los metodos usados anteriormente, el predictor-corrector con la matriz reducida. Si en este caso, a diferencia del sistema completo, la matriz del lado derecho del sistema no tiene inversa numéricamente; entonces, el metodo se detien por completo.

4. Funciones Prueba

4.1. Programa quadprog.m

En Matlab existe el programa quadprog.m el cual usa un método de predicción-corrección en cada iteración y genera el punto inicial. Éste otorga los siguientes resultados:

Problema	n	m	iter	$f(x^*)$
AFIRO	51	27	11	2.0082e + 05
CAPRI	496	271	34	9.3979e + 07
SC105	163	105	9	1.7220e + 05
BOEING1	726	351	28	NOCONV
GROW7	301	140	9	-88.3601
SCTAP1	660	330	12	1.4453e + 04

4.2. Programa qpintpointpc.m Opción 1

Problema	n	m	iter	$f(x^*)$
AFIRO	51	27	27	2.0082e + 05
CAPRI	496	271	100	1.02884e + 07
SC105	163	105	25	1.77199e + 05*
BOEING1	726	351	100	1.796279e + 08
GROW7	301	140	24	-88.3600
SCTAP1	660	330	54	1.4453 + 04

4.3. Programa *qpintpointpc.m* Opción 2

Problema	n	m	iter	$f(x^*)$
AFIRO	51	27	27	2.0082e + 05
CAPRI	496	271	100	1.0288e + 07
SC105	163	105	25	1.77199e + 05*
BOEING1	726	351	100	1.7963e + 08
GROW7	301	140	22	-88.3600
SCTAP1	660	330	55	1.4453e + 04

4.4. Programa *qpintpointpc.m* Opción 3

Problema	n	m	iter	$f(x^*)$
AFIRO	51	27	19	2.0082e + 05
CAPRI	496	271	100	1.02759e + 07
SC105	163	105	15	1.77199e + 05
BOEING1	726	351	100	1.809962e + 08
GROW7	301	140	15	-88.3601
SCTAP1	660	330	100	1.3684e + 04

4.5. Programa *qpintpointpc.m* Opción 4

Problema	n	m	iter	$f(x^*)$
AFIRO	51	27	2	5.8808e + 05
CAPRI	496	271	1	1.0139e + 07
SC105	163	105	3	3.127691e + 06
BOEING1	726	351	2	2.1797e + 08
GROW7	301	140	3	-51.8622
SCTAP1	660	330	100	2.0389e + 04

5. Conclusiones

Recalcaremos las observaciones principales sobre los distintos resultados obtenidos en las seis bases de datos.

En el caso de AFIRO, el valor óptimo $2.00823e+05$ se obtuvo en las opciones uno, dos y tres. En la opción cuatro, la matriz reducida en la segunda iteración era singular numéricamente, por lo que se dejó de iterar y no encontró el valor óptimo. La opción tres tuvo el mínimo de iteraciones, 19, mientras que la opción uno fue el que menos tardó en converger.

Prosiguiendo en la base de datos CAPRI, ninguna de las opciones converge en cien iteraciones, que fue el máximo permitido, y todas llegan a un valor similar. La opción cuatro encuentra que la matriz reducida es singular numéricamente por lo que se cancela el proceso de iteración. Probablemente se llegaría al óptimo al que llega quadprog.m si permitimos un número mayor de iteraciones.

Continuando con SC105, en las primeras tres opciones se llega al valor óptimo y coincide con el quadprog.m. En la opción cuatro nuevamente encuentra la matriz reducida que es singular numéricamente, por lo que se deja el proceso iterativo desde la tercera iteración. La opción tres nuevamente tuvo el mínimo de iteraciones con 15, así como el tiempo mínimo.

En el caso de BOEING1, ninguna de las opciones converge en cien iteraciones, que fue el máximo permitido, y todas llegan a un valor similar. La opción cuatro encuentra que la matriz reducida es singular numéricamente por lo que se cancela el proceso en la segunda iteración.

Terminando con GROW7, las primeras tres opciones llegan al mismo valor óptimo, que coincide con el quadprog.m. La opción cuatro en la tercera iteración encuentra que la matriz reducida es singular por lo que se cancela el proceso de iteración. La opción 3 tuvo el mínimo de iteraciones, 15, y obtuvo el menor tiempo de convergencia.

Finalmente con SCTAP1, las primeras tres opciones llegan al valor óptimo que coincide con el quadprog.m. La opción cuatro no converge en cien iteraciones, probablemente converga si permitimos un número mayor de iteraciones. La opción uno fue la que le tomó menos iteraciones en llegar al óptimo y también obtuvo el menor tiempo.