

Notas de Optimización II.
Unidad 3

Víctor Muñoz Sánchez

5 de enero de 2006

Capítulo 1

Métodos de Punto Interior para Programación Lineal

Estos métodos surgen a mediados de la década de los 80's, y son útiles para problemas lineales de gran escala. Están basados en el método de Newton.

Consideremos el problema de PL en forma estándar:

$$\text{mín } c^T x, \text{ subject to } Ax = b, x \geq 0 \quad (1.1)$$

donde c y x son vectores en \mathbb{R}^n , b es un vector en \mathbb{R}^m y A es una matriz de $m \times n$.

Los algoritmos de PI se caracterizan por requerir que todas las iteraciones satisfagan las restricciones de desigualdad en forma estricta (de ahí el nombre de *punto interior*). El método Simplex actúa alrededor de la frontera del polítopo factible, probando una secuencia de vértices hasta que encuentra el óptimo. Los métodos de PI se aproximan a la frontera del conjunto factible solamente en el límite. Pueden aproximar la solución ya sea desde el interior o el exterior de la región factible, pero no en la frontera de esta región.

1.1. Métodos Primal-Dual

El problema dual para (1.1) es

$$\text{máx } b^T \lambda, \text{ subject to } A^T \lambda + s = c, s \geq 0 \quad (1.2)$$

donde $\lambda \in \mathbb{R}^m$ y $s \in \mathbb{R}^n$. Las condiciones de Karush-Kuhn-Tucker (KKT) que caracterizan las soluciones para el problema dual son:

$$A^T \lambda + s = c \quad (1.3)$$

$$Ax = b \quad (1.4)$$

$$x^T s = 0 \quad (1.5)$$

$$(x, s) \geq 0 \quad (1.6)$$

Los métodos primal-dual encuentran soluciones (x^*, λ^*, s^*) de este sistema aplicando variantes del método de Newton para las tres igualdades en (1.3) y modificando las direcciones de búsqueda y tamaños de paso tales que las desigualdades $(x, s) \geq 0$ se satisfacen *estrictamente* en cada iteración.

Recordemos el método de Newton, que busca resolver:

$$\text{mín } f(x), \quad x \in \mathbb{R}^n$$

si $f(x)$ es convexa, equivale a resolver

$$\nabla f(x) = 0$$

usando Newton:

$$x^{k+1} = x^k + \alpha p$$

con p como solución de

$$\nabla^2 f(x^k) p^k = -\nabla f(x^k)$$

Entonces puede verse el método de optimización de Newton como un método de solución de sistemas no lineales.

Las ecuaciones (1.3) pueden reescribirse en la siguiente forma (sistema KKT):

$$F(x, \lambda, s) = \begin{pmatrix} A^T \lambda + s - c \\ Ax - b \\ X S e \end{pmatrix} = 0 \quad (1.7)$$

$$(x, s) \geq 0 \quad (1.8)$$

donde

$$X = \text{diag}(x_1, x_2, \dots, x_n), \quad S = \text{diag}(s_1, s_2, \dots, s_n) \quad (1.9)$$

y $e = (1, 1, \dots, 1)^T$. Entonces, los métodos primal-dual generan iteraciones (x^k, λ^k, s^k) que satisfacen estrictamente las restricciones de frontera (1.8). Definamos los conjuntos primal-dual *factible* \mathcal{F} y *estrictamente factible* \mathcal{F}^o como

$$\mathcal{F} = \{(x, \lambda, s) | Ax = b, A^T \lambda + s = c, (x, s) \geq 0\} \quad (1.10)$$

$$\mathcal{F}^o = \{(x, \lambda, s) | Ax = b, A^T \lambda + s = c, (x, s) > 0\} \quad (1.11)$$

así, podemos escribir la condición de factibilidad estricta como $(x^k, \lambda^k, s^k) \in \mathcal{F}^o$.

Como la mayoría de los algoritmos iterativos de optimización, los métodos primal-dual de PI tienen un procedimiento para determinar las direcciones de descenso y el tamaño de paso. Esto se realiza siguiendo el método de Newton, el cual forma un modelo lineal para F alrededor del punto actual y obtiene la dirección de búsqueda $(\Delta x, \Delta \lambda, \Delta s)$ resolviendo el sistema de ecuaciones lineales que se muestra:

$$J(x, \lambda, s) \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{pmatrix} = -F(x, \lambda, s)$$

donde J es el Jacobiano de F . Si $(x^k, \lambda^k, s^k) \in \mathcal{F}^o$, las ecuaciones de Newton se vuelven:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XSe \end{pmatrix} \quad (1.12)$$

y la nueva iteración será $(x, \lambda, s) + \alpha(\Delta x, \Delta \lambda, \Delta s)$, pero que no viole la restricción (1.8).

Un problema es encontrar un punto inicial estrictamente factible, pero el algoritmo anterior puede modificarse en la siguiente forma. Definamos los residuales para las dos ecuaciones lineales mediante

$$r_b = Ax - b, \quad r_c = A^T \lambda + s - c \quad (1.13)$$

y la ecuación de paso queda como

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \\ -XSe + \sigma \mu e \end{pmatrix} (x, s) \geq 0 \quad (1.14)$$

y esta dirección de búsqueda es aún un paso de Newton que trata de corregir la no factibilidad en las ecuaciones de igualdad en un solo paso. Puede verse que estos residuales se vuelven cero en las primeras iteraciones del algoritmo y las subsecuentes iteraciones permanecen estrictamente factibles. A este procedimiento se le llama *métodos de punto interior infactibles*.

Nótese que si tengo un $(x^0, \lambda^0, s^0) \in \mathcal{F}$ pero no son óptimos y además se satisface $x_i s_i \approx 0$, entonces una solución trivial al sistema de Newton es $(x^0, \lambda^0, s^0) = 0$ y no puede avanzar. La solución es modificar el vector XSe de forma tal que se garantice que no sea cero:

$$\begin{pmatrix} -r_c \\ -r_b \\ -XSe + e\tau \end{pmatrix}$$

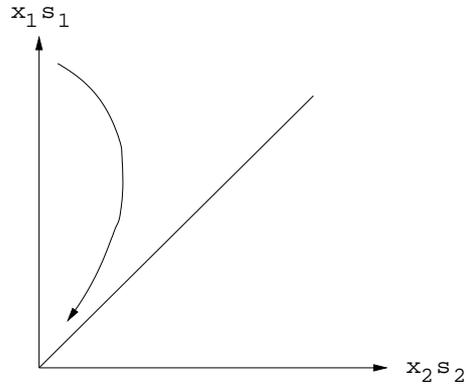
con $\tau \rightarrow 0$ para k grandes. Una opción para elegir τ es $\tau = 0,99^k$. Una mejor opción es elegirla dependiendo de los productos $s_i x_i$:

$$\tau = \sigma \mu$$

con $\sigma \in [0, 1]$ como *parámetro de centrado* y μ como *medida de dualidad* definida por

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i s_i = \frac{x^T s}{n} \quad (1.15)$$

Si $\sigma_i = 1$, entonces $s_i x_i = \mu$ y el método sesga la iteración al interior del octante. Por ejemplo, para $i = 2$, el comportamiento sería como se muestra en la figura 1.1.



Con todo esto, podemos expresar un esquema de un método primal-dual de PI, que se muestra en el algoritmo 1.

1.2. Resolución del sistema de Newton

Del tercer renglón de (1.14):

$$S\Delta x + X\Delta s = -XSe + \sigma \mu e$$

luego,

$$\Delta s = -Se + \sigma \mu X^{-1}e - X^{-1}S\Delta x$$

obteniéndose

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ -s + \sigma \mu X^{-1}e - X^{-1}S\Delta x \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \end{pmatrix}$$

Algorithm 1 Primal-Dual

- 1: Given $(x^0, \lambda^0, s^0) \in \mathcal{F}^o$
- 2: **for** $k = 0, 1, 2 \dots$ **do**
- 3: Solve

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{pmatrix} \quad (1.16)$$

where $\sigma_k \in [0, 1]$ and $\mu_k = \frac{(x^k)^T s^k}{n}$

- 4: Set

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k (\Delta x^k, \Delta \lambda^k, \Delta s^k) \quad (1.17)$$

choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$

- 5: **end for**
-

Del primer renglón:

$$A^T \Delta \lambda + (-s + \sigma \mu X^{-1} e - X^{-1} S \Delta x) = -r_c$$

luego

$$A^T \Delta \lambda - X^{-1} S \Delta x = -r_c + s - \sigma \mu X^{-1} e$$

Definiendo

$$D \stackrel{\text{def}}{=} S^{-1/2} X^{1/2}$$

$$-D^{-2} = -X^{-1} S$$

y tenemos:

$$\begin{pmatrix} 0 & A \\ A^T & -D^{-2} \end{pmatrix} \begin{pmatrix} \Delta \lambda \\ \Delta x \end{pmatrix} = \begin{pmatrix} -r_b \\ -r_c + s - \sigma \mu X^{-1} e \end{pmatrix} \quad (1.18)$$

con

$$\Delta s = -s + \sigma \mu X^{-1} e - X^{-1} S^{-1} \Delta x$$

A este sistema se le conoce como *Sistema Aumentado*.

Como $X^{-1} S$ es diagonal y no singular, podemos resolver para Δx del segundo renglón de (1.18), obteniendo:

$$AD^2 A^T \Delta \lambda = -r_b + A(-S^{-1} X r_c + x - \sigma \mu S^{-1} e) \quad (1.19)$$

$$\Delta s = -r_c - A^T \Delta \lambda \quad (1.20)$$

$$\Delta x = -x + \sigma \mu S^{-1} e - S^{-1} X \Delta s \quad (1.21)$$

que se conocen como *Ecuaciones Normales*.

1.3. Métodos seguidores de la ruta (path-following)

Estos métodos restringen explícitamente las iteraciones a una vecindad de la ruta central \mathcal{C} y siguen \mathcal{C} para una solución del problema de PL. Previendo que las iteraciones se acerquen demasiado a la frontera del octante no negativo, se asegura que las direcciones de búsqueda calculadas en cada iteración hagan al menos alguna cantidad mínima de progreso hacia la solución.

Definamos las siguientes vecindades:

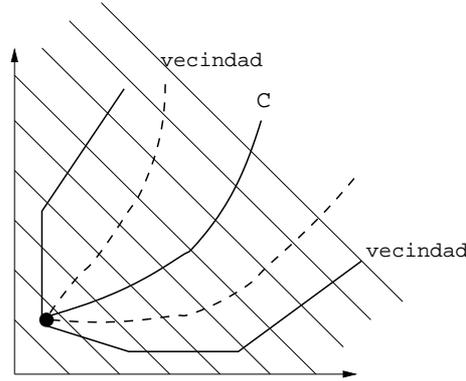
$$\mathcal{N}_2(\theta) = \{(x, \lambda, s) \in \mathcal{F}^o : \|XSe - \mu e\|_2 \leq \theta\mu\} \quad (1.22)$$

para un $\theta \in [0, 1)$

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, \lambda, s) \in \mathcal{F}^o | x_i s_i \geq \gamma\mu, \text{ para } i = 1, 2, \dots, n\} \quad (1.23)$$

para algún $\gamma \in (0, 1]$.

$\mathcal{N}_2(\theta)$ es más restrictiva, no importa si $\theta \approx 1$, algunos puntos de \mathcal{F}^o no estarán en \mathcal{N}_2 . A medida que se acerca uno al óptimo, $x_i s_i \rightarrow 0$ y se reduce \mathcal{N} , también $\mu \rightarrow 0$. La figura 1.3 muestra el comportamiento del algoritmo.



El algoritmo 2, muestra el procedimiento para un método seguidor de ruta, que puede hacer un rápido progreso por el uso de la vecindad amplia $\mathcal{N}_{-\infty}(\gamma)$ para γ cercana a 0. Aquí,

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) = (x^k, \lambda^k, s^k) + \alpha(\Delta x^k, \Delta \lambda^k, \Delta s^k)$$

Algorithm 2 Long-Step Path-Following

- 1: Given $\gamma, \sigma_{min}, \sigma_{max}$, con $\gamma \in (0, 1), 0 < \sigma_{min} < \sigma_{max} < 1$, and $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$
- 2: **for** $k = 0, 1, 2 \dots$ **do**
- 3: Choose $\sigma_k \in [\sigma_{min}, \sigma_{max}]$
- 4: Solve (1.16) to obtain $(\Delta x^k, \Delta \lambda^k, \Delta s^k)$
- 5: Choose α_k as the largest value of $\alpha \in [0, 1]$ such that

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma) \tag{1.24}$$

- 6: Set $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k(\alpha_k), \lambda^k(\alpha_k), s^k(\alpha_k))$

7: **end for**

Bibliografía

- [1] Jorge Nocedal y Stephen J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer.