O pacote computacional IPOPT

On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming

Andreas Wächter & Lorenz T. Biegler

Mathematical Programming 106, 25–57 (2006) | Cite this article

Problema a ser resolvido:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x_L \le x \le x_U, \end{aligned}$$

where $x_L \in [-\infty, \infty)^n$ and $x_U \in (-\infty, \infty]^n$, with $x_L^{(i)} \le x_U^{(i)}$,

f, c duas vezes continuamente diferenciáveis.

- restrições $g(x) \le 0 \longrightarrow g(x) + w = 0, \quad w \geq 0$

Apresentação da teoria para o problema simplificado

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x \ge 0. \end{aligned}$$

IPOPT implementa barreira logarítmica. Seus subproblemas são

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

onde $\mu \to 0^+$ (teoria vista em aula)

Resolve-se o subproblema acima por suas condições KKT:

veja as contas no anexo

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - z = 0 \\ c(x) = 0 \\ XZe - \mu e = 0, \end{cases} \qquad \left(e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)$$

onde, como visto em aula,

$$X = \operatorname{diag}(x_1, \ldots, x_n) = \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix}$$

(análogo para Z).

veja as contas no anexo

- Note que KKT para o problema original é o sistema acima com $\mu = 0$.

(a complementaridade $x_i z_i = 0$, $\forall i$, pode ser escrita como $XZe = 0$. Note que z_i é o multiplicador de Lagrange para $x_i \geq 0$).

Por isso queremos $\mu \to 0^+$.

Critério de parada

- Baseado no cumprimento aproximado das condições KKT do problema original.
- Medida de qualidade de KKT do subproblema

$$E_\mu(x, \lambda, z) := \max\left\{\frac{\|\nabla f(x) + \nabla c(x)\lambda - z\|_\infty}{s_d}, \|c(x)\|_\infty, \frac{\|XZe - \mu e\|_\infty}{s_c}\right\}$$

↳ $\|\nabla f(x) + \nabla c(x)\lambda - z\|_\infty$ é o resíduo da 1ª equação do sistema KKT.

↳ $\|c(x)\|_\infty$ é a medida de inviabilidade

↳ $\|XZe - \mu e\|_\infty$ é o resíduo da complementaridade.

↳ $s_d \geq 1$ e $s_c \geq 1$ são parâmetros de escalamento. Eles visam afrouxar a parada em problemas mal-condicionados.

↳ Veja que $\|c(x)\|_\infty$ não é escalada. Ou seja, obter um ponto viável é prioridade, IPOPT não flexibiliza isso.

Parada com sucesso

$$E_0(\tilde{x}_*, \tilde{\lambda}_*, \tilde{z}_*) \le \epsilon_{\text{tol}}$$

- $\varepsilon_{tol} > 0$: Tolerância fornecida pelo usário (padrão 10^{-8})
- $\mu = 0 \implies \approx$ KKT do problema original

Solução do subproblema

- Passos Newtonianos:

$$\begin{bmatrix} W_k & A_k & -I \\ A_k^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + A_k \lambda_k - z_k \\ c(x_k) \\ X_k Z_k e - \mu_j e \end{pmatrix} \qquad (*)$$

onde $W_k = \nabla^2_{xx} L(x_k, \lambda_k, z_k)$ e $A_k = \nabla c(x_k)$

- d = direção de Newton (visto em aula)
- IPOPT não resolve $(*)$ diretamente. Ao invés disso, resolve o seguinte sistema

equivalente:

$$\begin{bmatrix} W_k + \Sigma_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla \varphi_{\mu_j}(x_k) + A_k \lambda_k \\ c(x_k) \end{pmatrix} \quad (**)$$

onde $\Sigma_k := X_k^{-1} Z_k$. Com isso, a componente z da direção de Newton é calculada por

$$d_k^z = \mu_j X_k^{-1} e - z_k - \Sigma_k d_k^x$$

Veja que o sistema acima é menor e sua matriz é simétrica (W_k e Σ_k são simétricas). Isso facilita a resolução, pois existem boas técnicas para sistemas simétricos (Cholesky etc).

↳ Problema: se $A_k = \nabla c(x_k)$ não tiver posto completo (i.e., x_k não for regular), então a matriz do sistema $(**)$ pode ser singular, e uma solução pode não existir. Para contornar este problema, em

IPOPT resolve-se o sistema "perturbado"

$$\begin{bmatrix} W_k + \Sigma_k + \delta_w I & A_k \\ A_k^T & -\delta_c I \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla \varphi_{\mu_j}(x_k) + A_k \lambda_k \\ c(x_k) \end{pmatrix},$$

onde $\delta_w, \delta_c \geq 0$ são calculados a cada iteração de IPOPT. Este sistema sempre possui solução para certos $\delta_w, \delta_c \geq 0$, mesmo que $A_k = \nabla c(x_k)$ não tenha posto completo.

Novo iterando do método

Calculado (x_k, λ_k, z_k) e a direção de Newton d_k, o novo iterando consiste em um passo a partir de (x_k, λ_k, z_k) na direção d_k:

$$\begin{aligned} x_{k+1} &:= x_k + \alpha_k d_k^x \\ \lambda_{k+1} &:= \lambda_k + \alpha_k d_k^\lambda \\ z_{k+1} &:= z_k + \alpha_k^z d_k^z. \end{aligned}$$

onde os passos $\alpha_k, \alpha_k^z \in (0,1]$ são calculados para manter positividade de x e z.

Sendo $x_k > 0$ e $z_k > 0$, IPOPT garante os próximos $x_{k+1} > 0$ e $z_{k+1} > 0$ dando passos que mantenham uma fração de x_k e z_k:

$$\alpha_k^{\max} := \max\left\{\alpha \in (0,1] : x_k + \alpha d_k^x \geq (1-\tau_j)x_k\right\}$$
$$\alpha_k^z := \max\left\{\alpha \in (0,1] : z_k + \alpha d_k^z \geq (1-\tau_j)z_k\right\}$$

onde

$$\tau_j = \max\{\tau_{\min}, 1-\mu_j\}, \quad \tau_{\min} \in (0,1).$$

(pense que $j = k$)

Note que $(1-\tau_k)x_k > 0$ e que $1-\tau_k \to \tau_{\min}$ à medida que $\mu_k \to 0^+$.

A ideia é permitir $x_k \approx 0$ apenas no fim do método, quando $\mu_k \approx 0$.

Nota: IPOPT implementa muitas outras estratégias, não vistas em aula, que o tornam muito eficaz numericamente. A referência do pacote explica as principais.

Uma delas diz respeito ao tratamento de problemas com "interior vazio", ou seja, aqueles em que não existe x tal que

$$x_L < x < x_U$$

(relativo ao problema original

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x_L \leq x \leq x_U \end{aligned}$$

).

Antes de iniciar a resolução, por padrão IPOPT relaxa/alarga os limitantes x_L, x_U trocando-os da seguinte forma:

$$x_L^{(i)} \leftarrow x_L^{(i)} - \epsilon_{\text{tol}} \max\{1, |x_L^{(i)}|\}$$

onde $\varepsilon_{tol} > 0$ é a precisão de parada. Note que o lado direito é menor que x_L. Estratégia similar é usada para aumentar x_U.

↳ o exercício 4 da lista 2 discute essa estratégia

ANEXO

$SP(\mu)$: $\min_{x \in \mathbb{R}^n} \quad \varphi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \ln(x^{(i)})$

s.t. $c(x) = 0$

KKT do subproblema:

$$\nabla f(x) - \mu \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{bmatrix} + \nabla c(x)\lambda = 0,$$

$$c(x) = 0.$$

Definimos $z_i = \frac{\mu}{x_i}$, $\forall i$. Assim,

$$\nabla f(x) + \nabla c(x)\lambda - z = 0$$

$$c(x) = 0$$

$$x_i z_i - \mu = 0, \forall i$$

Notação: $X = \operatorname{diag}(x_1, \ldots, x_n) = \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix}$

$Z = \operatorname{diag}(z_1, \ldots, z_n) = \begin{bmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{bmatrix}$

Note que $XZe - \mu e = 0$, onde e é

o vetor de 1's, temos

$$XZe - \mu e = \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix} \begin{bmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - \mu \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} - \mu \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 z_1 \\ \vdots \\ x_n z_n \end{bmatrix} - \mu \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 z_1 - \mu \\ \vdots \\ x_n z_n - \mu \end{bmatrix}$$

Assim, $XZe - \mu e = 0 \iff x_i z_i - \mu = 0, \forall i$.

Daí, KKT do subproblema fica

$$\boxed{\begin{aligned} \nabla f(x) + \nabla c(x)\lambda - z &= 0 \\ c(x) &= 0 \\ XZe - \mu e &= 0 \end{aligned}}$$

(sistema no artigo de referência)

No método, $x_k > 0$, $\forall k$. Consequentemente,

$(z_k)_i = \dfrac{\mu_k}{(x_k)_i} > 0$, $\forall k, i$. $(\mu_k > 0)$.

KKT do problema original

$$\min_{x\in\mathbb{R}^n} \; f(x)$$
$$\text{s.t.} \; c(x) = 0$$
$$x \geq 0. \leftarrow -x \leq 0$$

$$\nabla f(x) + \nabla c(x)\lambda - z = 0$$

(z é multip. associado à $-x \leq 0$)

$$c(x) = 0$$

garantido pelo método! $\leftarrow$ $\boxed{x \geq 0 \;,\; z \geq 0}$

$$x_i z_i = 0 \;, \forall i$$

$(\mu = 0)$

Passo de Newton para o subproblema

$$\min_{x\in\mathbb{R}^n} \; \varphi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \ln(x^{(i)})$$
$$\text{s.t.} \; c(x) = 0$$

KKT:
$$\nabla f(x) + \nabla c(x)\lambda - z = 0$$
$$c(x) = 0$$
$$XZe - \mu e = 0$$

$$F(x,\lambda,z) = \begin{bmatrix} \nabla f(x) + \nabla c(x)\lambda - z \\ c(x) \\ XZe - \mu e \end{bmatrix}$$

<u>Newton</u>: $(x^{k+1}, \lambda^{k+1}, z^{k+1}) = (x^k, \lambda^k, z^k) + \alpha d^k$

onde d^k é solução do sistema

$$F'(x^k, \lambda^k, z^k)\, d = -F(x^k, \lambda^k, z^k)$$

$$F'(x,\lambda,z) = \begin{bmatrix} \nabla^2 f(x) + \sum_{i=1}^{m} \nabla^2 c_i(x)\lambda_i & \nabla c(x) & -I \\ \nabla c(x)^t & 0 & 0 \\ Z & 0 & X \end{bmatrix}$$

$$\nabla^2_{xx} L(x,\lambda,z) = W$$

$$A = \nabla c(x)$$

(matriz do sistema no artigo de referência)

Executando pacote IPOPT em um exemplo

```
julia> using NLPModelsIpopt
```

```
julia> include("prob.jl")
```

```
julia> println(P)
Min (x[1] - 2.0) ^ 2.0 + (x[2] - 1.0) ^ 2.0
Subject to
 x[1] + x[2] ≤ 2.0
 x[1] ≥ -10.0
 x[2] ≥ -10.0
 x[1] ≤ 10.0
 x[2] ≤ 10.0
 (x[1] ^ 2.0 - x[2]) - 0.0 ≤ 0
```

$$
\begin{aligned}
\text{minimizar} \quad & f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \\
\text{sujeito a} \quad & g_1(x) = x_1 + x_2 - 2 \leq 0 \quad \checkmark \\
& g_2(x) = x_1^2 - x_2 \leq 0. \quad \checkmark
\end{aligned}
$$

Minimizador: $x^* = (1,1)$, $f(x^*) = 1$.

```
julia> saida = ipopt(nlp,tol=1e-10)
```

F.O. — $\|c(x)\|_\infty$ — $\|\nabla f \cdots\|_\infty$ — $\|d\|$ (direção de Newton)

```
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.0000000e+00 0.00e+00 2.94e-01  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  1.1438333e+00 0.00e+00 2.41e-03  -1.0 9.28e-02    -  9.93e-01 1.00e+00f  1
   2  1.0467454e+00 0.00e+00 3.86e-03  -1.7 6.90e-02    -  1.00e+00 1.00e+00h  1
   3  1.0018560e+00 0.00e+00 6.79e-04  -3.8 3.46e-02    -  1.00e+00 1.00e+00h  1
   4  1.0000089e+00 0.00e+00 1.92e-07  -5.7 2.27e-03    -  1.00e+00 1.00e+00h  1
   5  9.9999999e-01 0.00e+00 2.21e-12  -8.6 1.16e-05    -  1.00e+00 1.00e+00h  1

Number of Iterations....: 5
```

$\log(\mu)$ — passos α.

```
                                   (scaled)                 (unscaled)
Objective...............:   9.9999998516736233e-01    9.9999998516736233e-01
Dual infeasibility......:   2.2134336861702505e-12    2.2134336861702505e-12
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   2.6740268467060290e-09    2.6740268467060290e-09
Overall NLP error.......:   2.6740268467060290e-09    2.6740268467060290e-09
```

F.O

$\|XZe - \mu e\|_\infty$

$E_0(x, \lambda, z)$

$\|c(x)\|_\infty$

$\| \nabla f(x) + \nabla c(x) \lambda - z \|_\infty$

Escalamento do problema (saída "scaled")

$$\begin{aligned} \min \quad & w_f f(x) \\ \text{s.a.} \quad & w_1 c_1(x) = 0 \\ & w_2 c_2(x) = 0 \\ & \vdots \\ & w_m c_m(x) = 0 \\ & x \geq 0 \end{aligned}$$

$w_1 \to 10^{-6}$

$w_m \to 1$

$10^6 x_1 + 10^{10} x_2 = 0$

$\nabla : \begin{bmatrix} 10^6 \\ 10^{10} \end{bmatrix}$

$x_1 - 0{,}01 x_2 = 0$

$\nabla : \begin{bmatrix} 1 \\ -0{,}01 \end{bmatrix}$

$$\nabla f + \underbrace{\nabla c_1 \lambda_1}_{\gg 1} + \underbrace{\nabla c_2 \lambda_2}_{\ll 1} \cdots - z = 0$$

$\underbrace{w_f, w_1, \ldots, w_m}_{>0}$: fatores de escalamento, visam reescrever o problema tentando colocar os dados na mesma magnitude.

Suponha que queremos $\|\nabla f + \nabla c(x)\lambda - z\|_\infty \leq 10^{-6}$.
Se escalamos o problema, passamos a testar
$\|w_f \nabla f + w \nabla c(x)\lambda - z\|_\infty \leq 10^{-6}$. $\neq$

```
julia> saida.objective
0.9999999800181822

julia> saida.solution
2-element Vector{Float64}:
 1.000000009990909
 1.0000000099954545
```