

Métodos de Pontos Interiores Aplicados ao Problema de Regressão pela Norma L_p ¹

A.R.L. OLIVEIRA², D.R. CANTANE³, Instituto de Matemática, Estatística e de Ciência da Computação, UNICAMP, 13081-970 Campinas, SP, Brasil.

Resumo. Os métodos de pontos interiores barreira logarítmica são desenvolvidos para o problema de regressão pela norma L_p e a estrutura matricial resultante é explorada objetivando uma implementação eficiente. Alguns conceitos dos métodos de pontos interiores são apresentados e um método de convergência quadrática existente é descrito. As implementações dos métodos de pontos interiores desenvolvidos são comparadas com o método já existente obtendo melhor desempenho computacional para problemas de grande porte.

1. Introdução

O método IRLS *iteratively reweighted least-squares* [6] foi, por muito tempo, a única alternativa prática para a resolução do problema de regressão pela norma L_p . Mais recentemente este método foi aperfeiçoado, no que diz respeito a robustez, através da inclusão de uma busca linear [4]. No mesmo trabalho, foi também proposto um novo método que apresenta características similares aos métodos de pontos interiores. Este método apresentou resultados computacionais superiores ao IRLS.

Ambos métodos apresentados em [4] têm uma importante desvantagem: a busca linear é computacionalmente cara. Isto leva a acreditar que os métodos de pontos interiores aplicados a este problema podem obter resultados computacionais superiores, repetindo o desempenho obtido na minimização pelas normas L_1 e L_∞ em [8, 9], respectivamente.

Um terceiro método, desenvolvido em [2], utiliza a relaxação de coluna para calcular a solução da norma L_p de um sistema de equações lineares inconsistentes. Não realizamos experimentos computacionais com este método uma vez que somente uma variável é atualizada a cada iteração, devendo convergir muito lentamente.

2. O Problema de Regressão pela Norma L_p

O problema de regressão

$$\min_{x \in \mathbb{R}^n} \|A^t x - b\|_p^p, \quad (2.1)$$

¹Este trabalho tem apoio parcial da FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo) e CNPq (Conselho Nacional Brasileiro de Desenvolvimento Científico e Tecnológico).

²aurelio@ime.unicamp.br

³dcantane@ime.unicamp.br

onde $A = [a_1, \dots, a_m] \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$ e $m > n$, tem inúmeras aplicações em diversas áreas de ciências e engenharias. As normas mais utilizadas são as normas L_2 , muito popular, entre outros motivos, por permitir uma solução direta; L_1 , que permite diminuir o efeito de pontos discrepantes e L_∞ que garante proteção contra o pior caso. Os dois últimos problemas podem ser formulados por programação linear e os métodos de pontos interiores aplicados a estes problemas permitem a exploração da estrutura matricial do problema de forma bastante eficiente [8, 9].

O problema de regressão L_p pode ser formulado da seguinte forma:

$$\begin{aligned} & \text{minimize } \|r\|_p^p & (2.2) \\ & \text{sujeito a } Ax + r = b, \end{aligned}$$

onde $1 < p < 2$. Este problema pode combinar as propriedades de regressão das normas 1 e 2 de forma apropriada para cada aplicação.

Definindo $r = u - v$, podemos reescrever o problema (2.2) da seguinte forma:

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n (u_i + v_i)^p & (2.3) \\ & \text{sujeito a } Ax + u - v = b, \quad (u, v) \geq 0, \end{aligned}$$

pois sempre existe uma solução ótima tal que $u_i = 0$ ou $v_i = 0$. Com este modelo não é necessário trabalhar com valores absolutos e, portanto, temos uma função cuja primeira derivada é definida para qualquer ponto. Além disso, quando $p = 1$ temos o modelo de regressão L_1 resultando em um problema de otimização linear.

2.1. Método GNCS

Nesta seção, será apresentado um resumo do método desenvolvido em [4] para o problema de regressão pela norma L_p . O método referido como GNCS é um método de Newton globalizado que usa as condições de folgas complementares para o problema da norma L_p . O conteúdo e as notações desta seção estão de acordo com o artigo [4].

Dado o ponto inicial $r^0 = A^t x^0 - b$ com $|r^0| > 0$ e λ^0 .

Passo1: Calcular

$$\theta^k = (\eta^k e) ./ (\gamma |g^k| + \eta^k e),$$

onde $0 < \gamma < 1$, $e^t = [1, \dots, 1] \in \mathbb{R}^m$, $g^k = p(|r^k|)^{p-1} \sigma^k$ e

$$\eta^k = \max \left\{ \max \left\{ \frac{|D_r^k(g^k - \lambda^k)|}{\phi(r^0)} \right\}, \max \{ \max \{ |\lambda^k| - |g^k|, 0 \} \} \right\}.$$

Sejam $D_r^k = \text{diag}(|r^k|)$, $D_\theta^k = \text{diag}(|pg^k - (e - \theta^k) .* \lambda^k|)$.

Defina $D^k = (D_r^k (D_\theta^k)^{-1})^{1/2}$;

Passo2: Calcule a direção d^k por

$$\begin{cases} A (D^k)^{-2} r = 0, \\ A (D^k)^{-2} A^t dx^k = -AD^k g^k, \\ d^k = A^t dx^k; \end{cases}$$

Atualize λ^{k+1} : $\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k$.

Passo3: Calcule

$$\tau^k = \max \left(\tau, 1 - \frac{\eta^k}{\gamma + \eta^k} \right).$$

Use o procedimento de busca linear para encontrar α^k , descrito a seguir.

Atualize $r^{k+1} \leftarrow r^k + \alpha^k d^k$, $k \leftarrow k + 1$.

Vá para o passo 1.

2.1.1. Procedimento de Busca Linear

Dados τ^k , $\beta_f \in (0, 1)$, d^k , r^k , $\check{\alpha}^k$, $\rho_b > 0$ (p.ex. 10^6) e α_i^k definido por

$$J = \left\{ \alpha_i^k : \alpha_i^k = -\frac{r_i^k}{d_i^k}, r_i^k d_i^k < 0 \right\}.$$

Passo1: Seja $\alpha_*^k = \min(r^k + \alpha_i^k d^k)$ com $g(r^k + \alpha_*^k d^k)^t d^k \geq 0$. Se

$$\phi(r^{k+1}) \leq \phi(r^k) + \beta_f \alpha^k \nabla \phi(r^k)^t d^k, \quad (2.4)$$

onde $r^{k+1} = r^k + \alpha^k d^k$ é satisfeito com α_*^k , seja $\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k < \alpha_*^k \}$ e defina $\alpha^k \leftarrow \alpha_{\#}^k + \tau^k (\alpha_*^k - \alpha_{\#}^k)$ e retorna; caso contrário, continua;

Passo 2: Se (2.4) não é satisfeito com $\alpha^k = 1$, continua para o próximo passo.

Caso contrário, estabeleça

$$\alpha^k \leftarrow \begin{cases} 1, & \text{se } \min(|r^k + d^k|) > 0; \\ \alpha_{\#}^k + \tau^k (1 - \alpha_{\#}^k), & \text{caso contrário,} \end{cases}$$

onde $\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k < 1 \}$, retorna;

Passo 3: Seja

$$\alpha^k \leftarrow \begin{cases} \check{\alpha}^k, & \text{se } \min(|r^k + \check{\alpha}^k d^k|) > 0; \\ \alpha_{\#}^k + \tau^k (\check{\alpha}^k - \alpha_{\#}^k), & \text{caso contrário,} \end{cases}$$

onde $\alpha_{\#}^k \leftarrow \max \{ \alpha_i^k : 0 \leq \alpha_i^k < \check{\alpha}^k \}$ e $\check{\alpha}^k = -\frac{g^{k^t} d^k}{d^{k^t} \text{diag}(p(|r^k|)^{p-2}) d^k}$, retorna.

2.1.2. Critério de Convergência

$$\frac{|\phi(r^{k+1}) - \phi(r^k)|}{\phi(r^{k+1})} < \epsilon \quad \text{ou} \quad \eta^k < \epsilon.$$

3. Método Barreira Logarítmica

O problema (2.3) também pode ser escrito como

$$\begin{aligned} \min \quad & \sum_{i=1}^n (u_i + v_i)^p \\ \text{s.a.} \quad & Ax + u - v - b = 0, \quad (u, v) \geq 0. \end{aligned}$$

A função objetivo é denotada em termos de u e v por $\phi(u, v) = \sum_{i=1}^n (u_i + v_i)^p$,

o gradiente $\nabla\phi(u, v)$ é denotado por $g = \begin{bmatrix} g_u \\ g_v \end{bmatrix}$, onde $g_{u_i} = g_{v_i} = p(u_i + v_i)^{p-1}$ e $\nabla^2\phi = \begin{bmatrix} \nabla g_u \\ \nabla g_v \end{bmatrix}$, onde $\nabla g_{u_{ij}} = \nabla g_{v_{ij}} = \begin{cases} \frac{p(p-1)}{(u_i + v_i)^{2-p}}, & \text{se } i = j, \\ 0, & \text{se } i \neq j \end{cases}$

é uma matriz diagonal denotada por G .

Utilizando o Método Barreira Logarítmica [3, 11], temos

$$\begin{aligned} \min \quad & \sum_{i=1}^n (u_i + v_i)^p - \mu \sum_{i=1}^n \ln(u_i) - \mu \sum_{i=1}^n \ln(v_i) \\ \text{s.a.} \quad & Ax + u - v - b = 0, \end{aligned}$$

onde $\mu > 0$ é o parâmetro barreira ($\mu \rightarrow 0$). A Lagrangeana é dada por

$$L = \sum_{i=1}^n (u_i + v_i)^p - \mu \sum_{i=1}^n \ln(u_i) - \mu \sum_{i=1}^n \ln(v_i) + y^t(Ax + u - v - b),$$

onde y é o multiplicador de Lagrange.

Aplicando as condições de otimalidade [11], obtemos

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ g - \mu U^{-1} e + y \\ g - \mu V^{-1} e - y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.1)$$

onde U e V são matrizes diagonais cujos elementos não nulos são u e v respectivamente e $e = (1, 1, \dots, 1)^t$.

Reescrevendo as duas últimas equações de (3.1),

$$\underbrace{\nabla L}_{J(x,y,u,v)} = \begin{bmatrix} A^t y \\ Ax + u - v - b \\ U(g + y) \\ V(g - y) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e \\ \mu e \end{bmatrix}.$$

Utilizando o Método de Newton [11], chegamos a

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & I(g+y) + UG & UG \\ 0 & -V & VG & I(g-y) + VG \end{bmatrix} \begin{bmatrix} dx \\ dy \\ du \\ dv \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} -A^t y \\ -Ax - u + v + b \\ -U(g+y) + \mu e \\ -V(g-y) + \mu e \end{bmatrix} \quad (3.2)$$

Resolvendo (3.2), obtemos as direções dx, dy, du, dv . Através de eliminação de variáveis o sistema (3.2) se reduz a

$$\begin{aligned} dx &= (A^t DA)^{-1} r, \\ dy &= D \left\{ r_2 - Adx - D_u^{-1} r_3 + [D_v - VUG^2 D_u^{-1}]^{-1} (I + D_u^{-1} UG) (r_4 - D_u^{-1} VGr_3) \right\}, \\ dv &= [D_v - VUG^2 D_u^{-1}]^{-1} [r_4 + Vdy + D_u^{-1} (-VGr_3 + VUGdy)], \\ du &= r D_u^{-1} (r_3 - Udy - UGdv), \end{aligned}$$

onde

$$\begin{aligned} r_1 &= -A^t y, \\ r_2 &= -Ax - u + v + b, \\ r_3 &= -U(g + y) + \mu e, \\ r_4 &= -V(g - y) + \mu e, \\ D_u &= [I(g + y) + UG], \\ D_v &= [I(g - y) + VG], \\ D &= \left\{ D_u^{-1} U + [D_v - VUG^2 D_u^{-1}]^{-1} [V(I + D_u^{-1} UG)^2] \right\}^{-1}, \\ r &= \left\{ r_1 - A^t D \left[r_2 - D_u^{-1} r_3 + [D_v - VUG^2 D_u^{-1}]^{-1} (I + D_u^{-1} U) (r_4 - D_u^{-1} VGr_3) \right] \right\}. \end{aligned}$$

No cálculo do passo α , u e v devem permanecer estritamente positivos [11].

$$\alpha = \min \left\{ \tau \left(\min_{du_i < 0} -\frac{u_i}{du_i} \right), \tau \left(\min_{dv_i < 0} -\frac{v_i}{dv_i} \right), 1 \right\}, \text{ onde } \tau = 0.9995.$$

Conhecendo as direções e os passos, as variáveis podem ser atualizadas por

$$\begin{aligned} x^{k+1} &= x^k + \alpha dx, \quad y^{k+1} = y^k + \alpha dy, \\ u^{k+1} &= u^k + \alpha du, \quad v^{k+1} = v^k + \alpha dv. \end{aligned}$$

A atualização do parâmetro barreira é dada por

$$\mu^{k+1} = \frac{\mu^k}{\beta}, \quad \text{onde } \beta > 1.$$

3.1. Critério de Convergência

O critério de convergência é baseado nas condições de otimalidade (3.1):

$$N = \frac{\|\nabla L\|}{(1 + \|x\| + \|u\| + \|v\| + \|y\|)(2n)} \leq \epsilon.$$

Utilizamos também um critério de convergência baseado na diferença dos valores de N atual e da iteração anterior: $|N^{k+1} - N^k| \leq \epsilon_1$.

3.2. Pontos Iniciais

Os pontos iniciais são calculados baseados nas idéias desenvolvidas em [1],

$$x^0 = (A^t A)^{-1} A^t b, \quad r^0 = b - Ax^0, \quad y^0 = \frac{\kappa r^0}{\|r^0\|_\infty},$$

$$u_i^0 = \begin{cases} \frac{\lambda+1}{2} r_i^0 & \text{se } r_i^0 > 0, \\ -\frac{\lambda-1}{2} r_i^0 & \text{caso contrário,} \end{cases} \quad v_i^0 = \begin{cases} \frac{\lambda-1}{2} r_i^0 & \text{se } r_i^0 > 0, \\ -\frac{\lambda+1}{2} r_i^0 & \text{caso contrário.} \end{cases}$$

Em [1] u e v não são definidos, no entanto esta escolha foi proposta em [9] de tal forma que satisfaça a relação $u^0 + v^0 = |\lambda r^0|$. Assim, se λ é $O(1)$, ambos u^0 e v^0 são da mesma ordem de r^0 . Lembramos que, por definição, $r = u - v$. Estabelecemos $\kappa = 0,975$.

4. Método Predictor-Corretor

No método predictor-corrector [5, 7, 10], primeiramente tomamos um passo afim em que o parâmetro barreira $\mu = 0$. Assim, temos que resolver o sistema

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & I(g+y) + UG & UG \\ 0 & -V & VG & I(g-y) + VG \end{bmatrix} \begin{bmatrix} \bar{d}x \\ \bar{d}y \\ \bar{d}u \\ \bar{d}v \end{bmatrix} = \begin{bmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \bar{r}_3 \\ \bar{r}_4 \end{bmatrix}$$

$$= \begin{bmatrix} -A^t y \\ -Ax - u + v + b \\ -U(g+y) \\ -V(g-y) \end{bmatrix} \quad (4.1)$$

que é o mesmo sistema dado pela Equação (3.2) com $\mu = 0$.

Resolvendo o sistema (4.1) e através de eliminação de variáveis, obtemos as direções $\bar{d}x, \bar{d}y, \bar{d}u, \bar{d}v$, que são as mesmas direções de (3.2) com $\mu = 0$. Dessa forma, obtemos as direções afins do nosso problema. Agora, podemos impor um valor para o parâmetro barreira μ e a correção não linear para resolver o segundo sistema linear que é dado por

$$\begin{bmatrix} 0 & A^t & 0 & 0 \\ A & 0 & I & -I \\ 0 & U & I(g+y) + UG & UG \\ 0 & -V & VG & I(g-y) + VG \end{bmatrix} \begin{bmatrix} \hat{d}x \\ \hat{d}y \\ \hat{d}u \\ \hat{d}v \end{bmatrix} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \\ \hat{r}_4 \end{bmatrix}$$

$$= \begin{bmatrix} -A^t y \\ -Ax - u + v + b \\ -U(g+y) + \mu e - \bar{d}\bar{U}\bar{d}\bar{V}e \\ -V(g-y) + \mu e - \bar{d}\bar{U}\bar{d}\bar{V}e \end{bmatrix},$$

onde $\bar{d}\bar{U}$ e $\bar{d}\bar{V}$ são matrizes diagonais de $\bar{d}\bar{u}$ e $\bar{d}\bar{v}$ respectivamente (direções afins obtidas no primeiro sistema linear). Observe que temos a mesma matriz nos dois sistemas lineares, a diferença entre eles se encontra apenas nos vetores \hat{r}_3, \hat{r}_3 e \hat{r}_4, \hat{r}_4 . A fatoração da matriz não é afetada por μ , pois o mesmo aparece apenas nos vetores r_3 e r_4 . Assim, temos as direções

$$\begin{aligned}
\hat{d}x &= (A^t \bar{D}A)^{-1} \hat{r}, \\
\hat{d}y &= \bar{D} \left\{ \hat{r}_2 - A\hat{d}x - \bar{D}_u^{-1} \hat{r}_3 + [\bar{D}_v - VUG^2 \bar{D}_u^{-1}]^{-1} (I + \bar{D}_u^{-1}UG) (\hat{r}_4 - \bar{D}_u^{-1}VG\hat{r}_3) \right\}, \\
\hat{d}v &= [\bar{D}_v - VUG^2 \bar{D}_u^{-1}]^{-1} [\hat{r}_4 + V\hat{d}y + \bar{D}_u^{-1} (-VG\hat{r}_3 + VUG\hat{d}y)], \\
\hat{d}u &= \bar{D}_u^{-1} (\hat{r}_3 - U\hat{d}y - UG\hat{d}v), \\
\text{onde} \\
\hat{r}_1 &= -A^t y, \\
\hat{r}_2 &= -Ax - u + v + b, \\
\hat{r}_3 &= -U(g + y) + \mu e - \bar{d}U\bar{d}V e, \\
\hat{r}_4 &= -V(g - y) + \mu e - \bar{d}U\bar{d}V e, \\
\bar{D}_u &= [I(g + y) + UG], \\
\bar{D}_v &= [I(g - y) + VG], \\
\bar{D} &= \left\{ \bar{D}_u^{-1}U + [\bar{D}_v - VUG^2 \bar{D}_u^{-1}]^{-1} \left[V(I + \bar{D}_u^{-1}UG)^2 \right] \right\}^{-1}, \\
\hat{r} &= \left\{ \hat{r}_1 - A^t \bar{D} \left[\hat{r}_2 - \bar{D}_u^{-1} \hat{r}_3 + [\bar{D}_v - VUG^2 \bar{D}_u^{-1}]^{-1} (I + \bar{D}_u^{-1}UG) (\hat{r}_4 - \bar{D}_u^{-1}VG\hat{r}_3) \right] \right\}.
\end{aligned}$$

O cálculo do passo α e a atualização das variáveis são análogos ao método primal-dual barreira logarítmica.

4.1. Algumas Considerações

O critério de convergência e o ponto inicial são os mesmos utilizados no método de pontos interiores barreira logarítmica. Na maioria dos métodos de pontos interiores o passo mais caro em termos computacionais é a solução do sistema linear com uma matriz do tipo A^tDA , onde D é uma matriz diagonal. O sistema linear obtido em nosso trabalho é simétrico e definido positivo e assim pode ser resolvido pela fatoração de Cholesky. Obtemos, então, um sistema muito menor que o sistema original (2.1) e da mesma dimensão dos sistemas resolvidos pelo método do Y. Li [4]. No método preditor-corretor, apesar de resolvermos dois sistemas lineares em cada iteração, os cálculos utilizados para construir a matriz são efetuados uma única vez pois a matriz é a mesma para ambos sistemas lineares.

5. Resultados Computacionais

Aproximamos $f(z)$, calculado em $z = 0, \frac{1}{m}, \dots, 1$, por um polinômio de grau $n - 1$ tal que as normas L_p são minimizadas. As duas funções testes utilizadas são as mesmas funções utilizadas em [4] e são dadas por

$$f_1(z) = \sqrt{1+z}, \quad f_2(z) = e^z + \begin{cases} 5, & \text{se } 0, 1 < z < 0, 2, \\ 0, & \text{caso contrário.} \end{cases}$$

O método de pontos interiores primal-dual barreira logarítmica, referido como PD, assim como o método preditor-corretor, referido como PC e o método de convergência GNCS foram implementados em Matlab 5.3, com o sistema operacional

Windows Milenium, processador Intel Celeron 700MHz.

As Tabelas 1 e 2 mostram os resultados obtidos pelos programas métodos de pontos interiores barreira logarítmica, preditor-corretor e o método GNCS utilizando $f_1(z)$ e $f_2(z)$ respectivamente. Utilizamos o parâmetro barreira $\mu = 0,001$ e os critérios de parada $\epsilon = 10^{-5}$ e $\epsilon_1 = 10^{-3}$.

P	PD			PC			GNCS		
	It	Flops	FO	It	Flops	FO	It	Flops	FO
1	1	1948	0,0418	1	2351	0,0418	11	13759	0,0398
1,01	1	1948	0,0396	1	2351	0,0396	11	13831	0,0378
1,1	1	1948	0,0245	1	2351	0,0245	11	13831	0,0238
1,2	1	1952	0,0146	1	2359	0,0146	15	18831	0,0142
1,3	1	1948	0,0091	1	2351	0,0090	6	7511	0,0085
1,4	1	1948	0,0052	1	2351	0,0052	6	7511	0,0051
1,5	1	1948	0,0031	1	2351	0,0031	6	7511	0,0031
1,6	1	1952	0,0018	1	2359	0,0018	6	7491	0,0018
1,7	1	1948	0,0011	1	2351	0,0011	5	6261	0,0011
1,8	1	1948	6,5573e-4	1	2351	6,5572e-4	4	5031	6,5427e-4
1,9	1	1948	3,9242e-4	1	2351	3,9242e-4	4	5031	3,9208e-4

Tabela 1: Resultados computacionais utilizando a função $f_1(z)$.

P	PD			PC			GNCS		
	It	Flops	FO	It	Flops	FO	It	Flops	FO
1	2	3205	0,5566	2	4031	0,5558	7	8705	0,5280
1,01	2	3205	0,5417	2	4031	0,5409	7	8721	0,5145
1,1	2	3205	0,4240	2	4031	0,4238	10	12431	0,4081
1,2	2	3205	0,3242	2	4031	0,3239	8	9971	0,3162
1,3	2	3205	0,2485	2	4031	0,2477	13	16331	0,2457
1,4	2	3205	0,1923	2	4031	0,1903	10	12511	0,1902
1,5	1	1948	0,1476	1	2351	0,1485	11	13741	0,1473
1,6	1	1948	0,1170	1	2351	0,1172	9	11331	0,1141
1,7	1	1938	0,0910	2	4011	0,0907	8	10081	0,0884
1,8	2	3195	0,0694	2	4011	0,0693	6	7581	0,0685
1,9	2	3195	0,0534	2	4011	0,0534	4	5011	0,0532

Tabela 2: Resultados computacionais utilizando a função $f_2(z)$.

Utilizamos, agora, um problema de grande porte no qual estamos principalmente interessados em comparar o tempo de CPU em segundos e número de iterações necessários para a convergência dos métodos. O conjunto contém os dados “Prime Rate”(PR). O conjunto de dados compreende valores dos juros diários ao longo de 40 anos, finais-de-semanas e feriados não foram considerados, totalizando

10958 valores observados. Para aperfeiçoar a estabilidade numérica dos métodos, os dados (observados) foram normalizados no intervalo $[0, 1]$. Estes experimentos foram realizados em uma estação Sun Blade 100 com Matlab 6.0, assim, não foi possível estimar o número de flops. Utilizamos o parâmetro barreira $\mu = 0,01$ e os critérios de parada $\epsilon = 10^{-8}$, $\epsilon_1 = 10^{-4}$ e $\epsilon = 10^{-8}$, $\epsilon_1 = 10^{-6}$ respectivamente. Os resultados obtidos se encontram na Tabela 3.

P	PD		PC		GNCS	
	It	Tempo	It	Tempo	It	Tempo
1,1	1	817,3	1	936,5	3	2108,1
1,2	1	835,7	1	893,8	3	1499,3
1,3	1	695,2	1	622,9	3	1524,8
1,4	1	642,8	1	781,9	3	1851,1
1,5	1	512,8	1	1214,7	3	1475,3
1,6	2	1001,7	2	1775,6	3	1483,1
1,7	2	933,5	2	957,7	3	1502,4
1,8	2	1156,8	2	1009,1	3	1327,0
1,9	3	1341,6	3	1236,9	3	1268,1
2,0	3	1670,6	3	1334,5	2	870,8

P	PD		PC		GNCS	
	It	Tempo	It	Tempo	It	Tempo
1,1	7	2391,1	1	563,0	3	3026,1
1,2	7	2507,9	10	4546,1	3	2547,8
1,3	7	2921,6	8	3253,6	3	1982,7
1,4	7	3008,6	7	2974,8	5	3928,9
1,5	5	2158,6	5	2341,0	5	4377,7
1,6	5	2067,5	5	2317,1	5	4690,8
1,7	5	2151,5	5	2497,2	5	4484,3
1,8	5	2197,7	4	2207,7	4	3699,2
1,9	4	1713,7	4	2389,6	4	3850,9
2,0	4	1690,0	3	1712,1	3	2042,5

Tabela 3: Resultados computacionais utilizando o problema de grande porte.

6. Conclusões

Podemos verificar que os métodos de pontos interiores apresentam desempenho computacional superior ao método GNCS, tanto em relação à robustez quanto ao tempo computacional.

Os métodos de pontos interiores barreira logarítmica e preditor-corretor convergem para problemas de grande porte em menos iterações e com esforço por iteração muito menor que o método GNCS [4] utilizando o critério de convergência

(3.1.) e o $\epsilon_1 = 10^{-4}$. Utilizando o mesmo critério de convergência e $\epsilon_1 = 10^{-6}$, o GNCS obtém menor número de iterações mas como sua busca linear é muito cara o desempenho dos métodos de pontos interiores é superior. O método GNCS utilizando o critério de convergência (2.1.2.) proposto em [4] não converge mesmo após 100 iterações. Finalmente, o método primal-dual apresenta desempenho computacional ligeiramente superior ao método preditor-corretor.

Abstract. The logarithmic barrier interior point methods are developed for the regression norm L_p problem and the resultant matrix structure is exploited in order to have an efficient implementation. Some concepts of interior point methods are presented and an existing quadratic convergent method is described. Implementations of interior point methods developed are compared with the existing method obtaining a better computational performance for large scale problems.

Referências

- [1] T.F. Coleman e Y. Li, A globally and quadratically convergent affine scaling method for linear l_1 problems, *Math. Programming*, **56** (1992), 189-222.
- [2] A. Dax e B. Berkowitz, Column relaxation methods for least norm problems, *SIAM J. Sci. Stat. Comput.*, **11** (1990), 975-989.
- [3] A.S. El-Bakry, R.A. Tapia, T. Tsuchiya e Y. Zhang, On the formulation and the theory of the Newton interior-point method for nonlinear programming, *Journal of Optimization Theory and Applications*, **89** (1996), 507-541.
- [4] Y. Li, A globally convergent method for l_p problems, *SIAM J. Optimization*, **3** (1993), 609-629.
- [5] S. Mehrotra, On the implementation of a primal-dual interior point method, *SIAM Journal on Optimization*, **2** (1992), 575-601.
- [6] G. Merle e H. Späth, Computational experience with discrete l_p approximation, *Computing*, **12** (1974), 315-321.
- [7] R.D.C. Monteiro, I. Adler e M.G.C Resende, A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension, *Mathematics of Operations Research*, **15** (1990), 191-214.
- [8] A.R.L. Oliveira e C. Lyra, Interior point methods for the polynomial L_∞ fitting problems, *Internacional Transactions in Operational Research*, **11** (2004), 309-322.
- [9] A.R.L. Oliveira, M.A. Nascimento e C. Lyra, Efficient implementation and benchmark of interior point methods for the polynomial L_1 fitting problems, *Statistics & Data Analysis*, **35** (2000), 119-135.

- [10] D.F. Shanno e R.J. Vanderbei, Interior point methods for nonconvex nonlinear programming: Orderings and higher-order methods, *Mathematical Programming*, **87** (2000), 303-316.
- [11] S.J. Wright, “Primal-Dual Interior-Point Methods”, SIAM Publications, SIAM Philadelphia, PA, USA, 1996.

