Otimização e Algoritmos

Grupo 42:
89683, José Neves
89691, Leonardo Pedroso
100660, Gustavo Bakker

Professores:
João Xavier e Cláudia Soares

Instituto Superior Técnico

13 de dezembro de 2020

Conteúdo

Conteúdo

Problema de otimização - Controlar o Robot

$$\underset{(x,u)}{\text{minimizar}} \quad \sum_{k=1}^{K} \|Ex(\tau_k) - \omega_k\|_2^2 + \lambda \sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2^2$$

restrições:

$$\begin{aligned}
&x(0) = x_{inicial} \\
&x(T) = x_{final} \\
&\|u(t)\|_2 \leq U_{max}, para 0 \leq t \leq T-1 \\
&x(t+1) = Ax(t) + Bu(t), para 0 \leq t \leq T-1
\end{aligned}$$

Controlo do robot - 4 Wishes

$$\text{Wish 1}: x_{inicial} = (p_{inicial} 0),\ x_{final} = (p_{final} 0)$$

$$\text{Wish 2}: \|u(t)\|_2 \leq Umax, para 0 \leq t \leq T-1$$

$$\text{Wish 3}: p(\tau_k) \simeq \omega_k, para 1 \leq k \leq K$$

$$\text{Wish 4}: u(t) = u(t-1) t \in 1, ..., T-1$$

Controlo do robot - 4 Wishes

Define o peso do regularizador (wish 4)

$$\underset{(x,u)}{\text{minimizar}} \quad \overbrace{\sum_{k=1}^{K} \|Ex(\tau_k) - \omega_k\|_2^2}^{\text{Wish 3}} + \lambda \overbrace{\sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_2^2}^{\text{Wish 4}}$$

restrições:

$$\left.\begin{array}{l} x(0) = x_{inicial} \\ x(T) = x_{final} \end{array}\right\rbrace \text{Wish 1}$$

$$\boxed{\|u(t)\|_2 \leq U_{max}, \qquad para\ 0 \leq t \leq T-1} \ \text{Wish 2}$$

$$x(t+1) = Ax(t) + Bu(t), \qquad para\ 0 \leq t \leq T-1$$

Figura 1: Identificação dos 4 whishes e parâmetro λ

Comparação de resultados para ℓ_2^2, ℓ_2 e ℓ_1

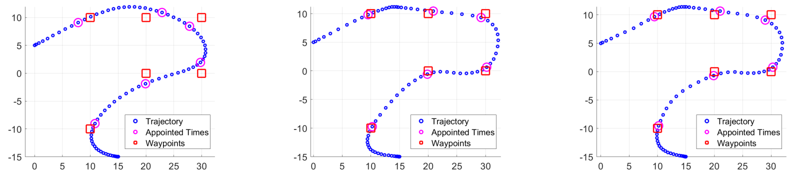
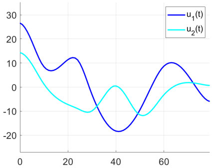
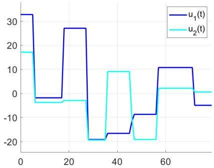


Figura 2: Trajetória para $\lambda = 10^{-1}$

Comparação de resultados para ℓ_2^2, ℓ_2 e ℓ_1





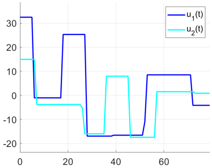


Figura 3: Sinal de controlo para $\lambda = 10^{-1}$

Comparação de resultados para ℓ_2^2, ℓ_2 e ℓ_1

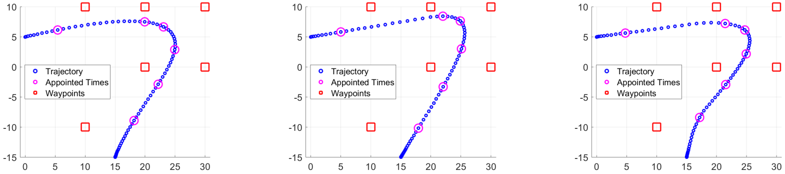


Figura 4: Trajetória para $\lambda = 10^1$

Comparação de resultados para ℓ_2^2, ℓ_2 e ℓ_1

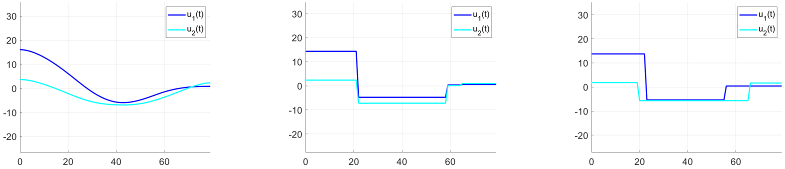


Figura 5: Sinal de controlo para $\lambda = 10^1$

Análise da evolução de cada norma - Série de Taylor

ℓ_2^2

$$\begin{aligned} ||x||_2^2 &= ||a||_2^2 + D_x ||x||_2^2\Big|_a (x-a) + \mathcal{O}((x-a)^T(x-a)) \\ &= ||a||_2^2 + 2a^T(x-a) + \mathcal{O}((x-a)^T(x-a)) \end{aligned} \tag{1}$$

ℓ_2

$$\begin{aligned} ||x||_2 &= ||a||_2 + D_x ||x||_2\Big|_a (x-a) + \mathcal{O}((x-a)^T(x-a)) \\ &= ||a||_2 + \frac{a^T}{||a||_2}(x-a) + \mathcal{O}((x-a)^T(x-a)) \end{aligned} \tag{2}$$

ℓ_1

$$\begin{aligned} ||x||_1 &= ||a||_1 + D_x ||x||_1\Big|_a (x-a) + \mathcal{O}((x-a)^T(x-a)) \\ &= ||a||_1 + \left[\mathrm{sgn}(a_1), \ldots, \mathrm{sgn}(a_n)\right](x-a) + \mathcal{O}((x-a)^T(x-a)) \end{aligned} \tag{3}$$

Pontos importantes dos resultados obtidos

1. ℓ_2^2: derivada da norma do regularizador é nula na origem.
2. ℓ_2 e ℓ_1: derivada não é nula na origem
3. O regularizador ℓ_2^2 penaliza diferenças muito grandes, mas é benevolente quando as diferenças são pequenas.
4. ℓ_2^2: contínuo
5. ℓ_2 e ℓ_1: constante por troços.

Comparação dos diferentes regularizadores

	Control de sinal			Desvio médio		
λ	ℓ_2^2	ℓ_2	ℓ_1	ℓ_2^2	ℓ_2	ℓ_1
10^{-3}	79	7	11	0.1257	0.0075	0.0107
10^{-2}	79	7	11	0.8242	0.0747	0.1055
10^{-1}	79	8	14	2.1958	0.7021	0.8863
10^{0}	79	4	9	3.6826	2.8877	2.8734
10^{1}	79	3	4	5.6316	5.3689	5.4362
10^{2}	79	2	2	10.9041	12.5914	13.0273
10^{3}	79	1	2	15.3304	16.2266	16.0463

Tabela 1: Comparação dos diferentes valores de λ entre os 3 regularizadores.

Pontos importantes dos resultados obtidos

1. Para qualquer λ em ℓ_2^2, as mudanças do sinal de controlo são máximas (79), enquanto que para ℓ_2 e ℓ_1 o sinal de controlo varia.
2. para l_1 as componentes de sinal de contrlo sao regularizadas independentemente.
3. para ℓ_2 as componentes de sinal de controlo são regularizadas de forma acoplada. l_2^2 penaliza grandes diferenças. Sinal de controlo suave.
4. logo ℓ_2 tem menos mudanças de sinal de controlo que ℓ_1 e são simultâneas.
5. Para qualquer regularizador ℓ_2^2, ℓ_2, ℓ_1 verificou-se um aumento dos desvios médios com o aumento do valor de λ.

Comparação dos diferentes regularizadores

$$\sum_{t=1}^{T-1} \|u(t) - u(t-1)\|_1 = \sum_{t=1}^{T-1} |u_1(t) - u_1(t-1)| + \sum_{t=1}^{T-1} |u_2(t) - u_2(t-1)|$$

Cont.

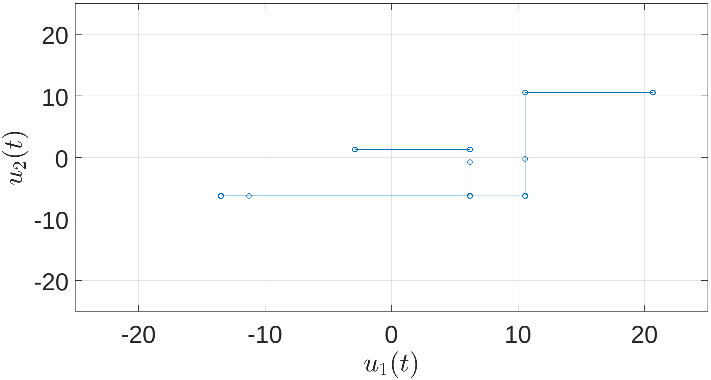


Figura 6: Representação do sinal de controlo ideal para $\lambda = 10^0$ com o ℓ_1 regularizador no plano.

Cont.

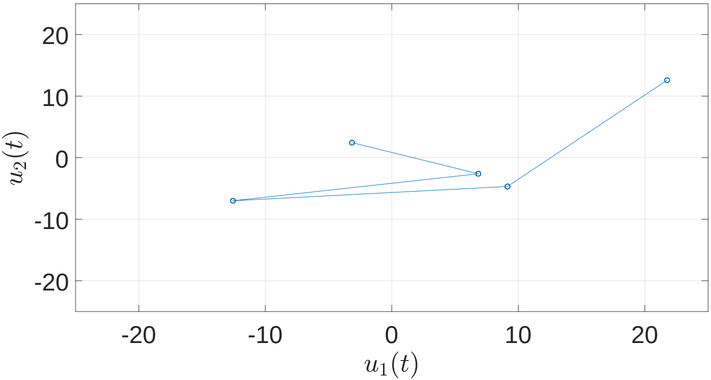


Figura 7: Representação do sinal de controlo ideal para $\lambda = 10^0$ com o ℓ_2 regularizador no plano.

Localizar um alvo em movimento

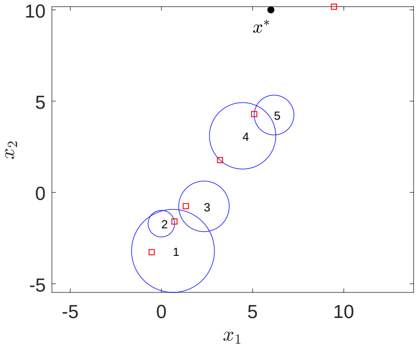


Figura 8: Deteção do ponto mais próximo de x*

Localizar um alvo em movimento

$$\begin{array}{ll} \underset{(p_0,v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimizar}} & ||p_0 + t^* v - x^*|| \\ \text{restrições} & ||p_0 + t_k v - c_k|| \leq R_k, \quad k = 1, ..., K, \end{array}$$

Menor rectângulo envolvente

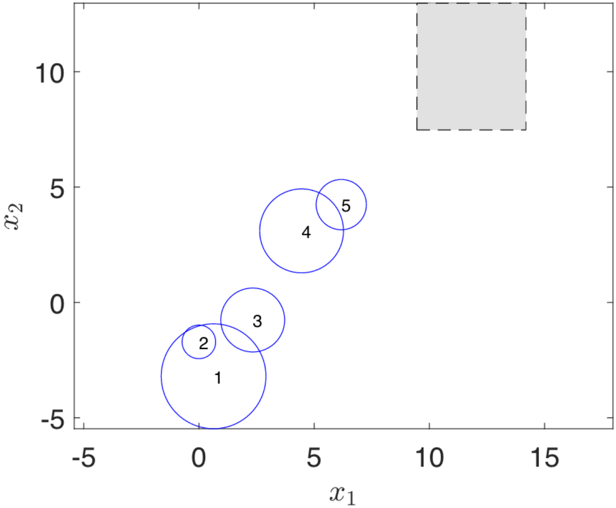


Figura 9: Método utilizado para encontrar os 4 pontos do rectângulo.

Menor rectângulo envolvente

$$\begin{array}{ll} \underset{(p_0,v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} & \begin{bmatrix}1 & 0\end{bmatrix}(p_0 + t^* v - x^*) \\ \text{subject to} & \|p_0 + t_k v - c_k\| \le R_k, \quad k = 1, ..., K. \end{array} \tag{4}$$

$$\begin{array}{ll} \underset{(p_0,v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} & \begin{bmatrix}-1 & 0\end{bmatrix}(p_0 + t^* v - x^*) \\ \text{subject to} & \|p_0 + t_k v - c_k\| \le R_k, \quad k = 1, ..., K, \end{array} \tag{5}$$

$$\begin{array}{ll} \underset{(p_0,v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} & \begin{bmatrix}0 & 1\end{bmatrix}(p_0 + t^* v - x^*) \\ \text{subject to} & \|p_0 + t_k v - c_k\| \le R_k, \quad k = 1, ..., K, \end{array} \tag{6}$$

$$\begin{array}{ll} \underset{(p_0,v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} & \begin{bmatrix}0 & -1\end{bmatrix}(p_0 + t^* v - x^*) \\ \text{subject to} & \|p_0 + t_k v - c_k\| \le R_k, \quad k = 1, ..., K. \end{array} \tag{7}$$

Menor região envolvente

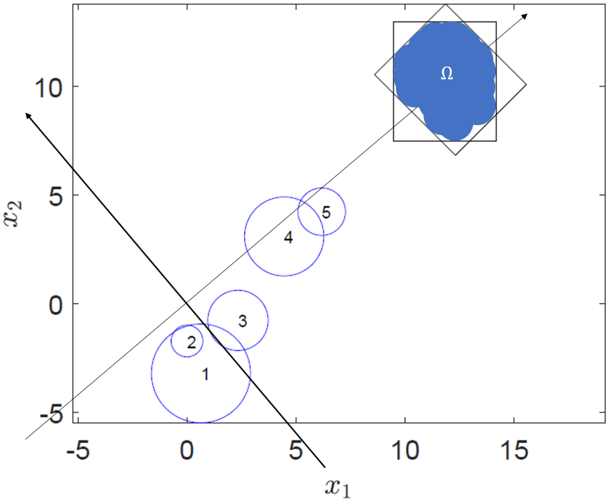


Figura 10: Figura-exemplo para explicação do método utilizado para aproximar o conjunto de soluções possíveis.

Menor região envolvente

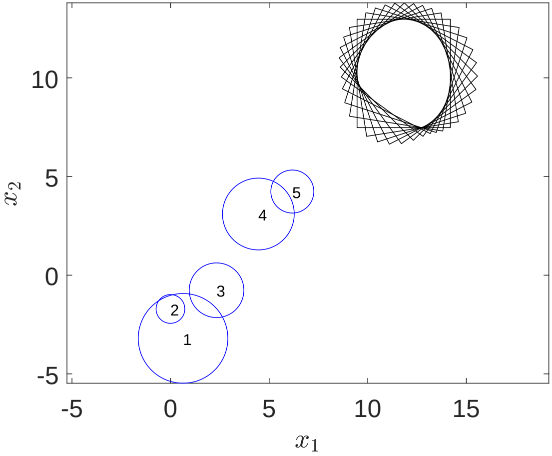


Figura 11: Menor polígono envolvente para $N = 10$ inclinações igualmente distanciadas.

Parte 1 - Menor região envolvente - convergência

Pretendemos resolver

$$\begin{array}{ll} \underset{\mathcal{D} \in \mathrm{R}^2}{\text{minimize}} & \mathrm{Area}(\mathcal{D}) \\ \text{subject to} & (\mathrm{p}_0 + t^* \mathrm{v}) \in \mathcal{D},\ \forall\ \mathrm{p}_0, \mathrm{v} \in \mathbb{R}^2 : \\ & ||\mathrm{p}_0 + t_k \mathrm{v} - \mathrm{c}_\mathrm{k}|| \leq R_k, \quad k = 1, ..., K\ . \end{array} \tag{8}$$

- É não-convexo
- É aparentemente intratável

Parte 1 - Menor região envolvente - convergência

Foi aproximado por N soluções de

$$
\begin{aligned}
&\underset{(\mathrm{p}_0, v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} && \begin{bmatrix}\cos\phi & \sin\phi\end{bmatrix}(\mathrm{p}_0 + t^*\mathrm{v}) \\
&\text{subject to} && ||\mathrm{p}_0 + t_k\mathrm{v} - \mathrm{c_k}|| \leq R_k, \quad k = 1, ..., K. \\
&\underset{(\mathrm{p}_0, \mathrm{v})\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} && \begin{bmatrix}-\cos\phi & -\sin\phi\end{bmatrix}(\mathrm{p}_0 + t^*\mathrm{v}) \\
&\text{subject to} && ||\mathrm{p}_0 + t_k\mathrm{v} - \mathrm{c_k}|| \leq R_k, \quad k = 1, ..., K, \\
&\underset{(\mathrm{p}_0, v)\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} && \begin{bmatrix}-\sin\phi & \cos\phi\end{bmatrix}(\mathrm{p}_0 + t^*\mathrm{v}) \\
&\text{subject to} && ||\mathrm{p}_0 + t_k\mathrm{v} - \mathrm{c_k}|| \leq R_k, \quad k = 1, ..., K, \\
&\underset{(\mathrm{p}_0, \mathrm{v})\in \mathrm{R}^2\times\mathrm{R}^2}{\text{minimize}} && \begin{bmatrix}\sin\phi & -\cos\phi\end{bmatrix}(\mathrm{p}_0 + t^*\mathrm{v}) \\
&\text{subject to} && ||\mathrm{p}_0 + t_k\mathrm{v} - \mathrm{c_k}|| \leq R_k, \quad k = 1, ..., K.
\end{aligned}
\tag{9}
$$

- Relaxação convexa

Parte 1 - Menor região envolvente - convergência

Theorem

Considere-se N retângulos, $\mathcal{R}_1, \ldots, \mathcal{R}_N$, cada um obtido da resolução de 4 problemas de otimização convexos (9)*, para $\phi \in \Phi$ com*

$$\Phi = \left\{ \phi \in \mathbb{R} : \phi = \frac{\pi}{2}((n-1)/N) \; ; n = 1, \ldots, N \right\} .$$

Então a interseção dos N retângulos, converge para a solução do problema não convexo (8) *à medida que $N \to \infty$, i.e.*

$$\lim_{N \to \infty} \cap_{i=1}^{N} \mathcal{R}_i = \mathcal{D} \,,$$

onde $\mathcal{D}$ é a solução de (8)*.*

Conteúdo

Problema de otimização

$$\underset{(s,r)\in \mathbb{R}^n\times \mathbb{R}}{\text{minimizar}} \quad \frac{1}{K}\sum_{k=1}^{K}\left(\log\left(1+\exp\left(s^T x_k - r\right)\right) - y_k\left(s^T x_k - r\right)\right)$$

Função objetivo

$$f(s,r) = \frac{1}{K}\sum_{k=1}^{K}\left(\log\left(1+\exp\left(s^T x_k - r\right)\right) - y_k\left(s^T x_k - r\right)\right)$$

Algoritmos

Lecionados

1. *Gradient descent*
   $\Rightarrow$ Algoritmo *line search* com $d_k = -\nabla f(x_k)$
   $\Rightarrow$ Para funções genéricas

2. Newton
   $\Rightarrow$ Algoritmo *line search* com $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
   $\Rightarrow$ Para funções convexas

3. Levenberg-Marquardt
   $\Rightarrow$ Parte 3

Convexidade

$$f(s,r) = \frac{1}{K}\sum_{k=1}^{K}\left(\log\left(1+\exp\left(s^T x_k - r\right)\right) - y_k\left(s^T x_k - r\right)\right)$$

$$f(s,r) = \sum_{k=1}^{K}\frac{1}{K}h_k(s,r) + \sum_{k=1}^{K}\frac{1}{K}l_k(s,r)$$

$$h_k(s,r) = \log\left(1+\exp\left(s^T x_k - r\right)\right)$$

$$l_k(s,r) = -y_k\left(s^T x_k - r\right)$$

$$h_k(s,r) = \left(t_k \circ q_k\right)(s,r)$$

$$l_k(s,r) = \begin{bmatrix} -y_k x_k \\ y_k \end{bmatrix}^T \begin{bmatrix} s \\ r \end{bmatrix} + 0$$

$$t_k(z) = \log(1+\exp(z))$$

$$q_k(s,r) = s^T x_k - r$$

$$q_k(s,r) = \begin{bmatrix} x_k \\ -1 \end{bmatrix}^T \begin{bmatrix} s \\ r \end{bmatrix} + 0$$

Figura 12: Decomposição da função objetivo para prova da sua convexidade.

Gradiente da função objetivo

$$\nabla f(s,r) = \frac{1}{K}\sum_{k=1}^{K}(\nabla h_k(s,r) + \nabla l_k(s,r))$$

$$\nabla h_k(s,r) = \nabla t_k(q_k(s,r))\nabla q_k(s,r)$$

$$\nabla l_k(s,r) = \begin{bmatrix} -y_k x_k \\ y_k \end{bmatrix} = -y_k \begin{bmatrix} x_k \\ -1 \end{bmatrix}$$

$$\nabla t_k(z) = \frac{dt_k}{dz}(z) = \frac{\exp(z)}{1+\exp(z)}$$

$$\nabla q_k(s,r) = \begin{bmatrix} x_k \\ -1 \end{bmatrix}$$

$$\nabla h_k(s,r) = \left(1 - \frac{1}{\exp\left(\begin{bmatrix} x_k \\ -1 \end{bmatrix}^T \begin{bmatrix} s \\ r \end{bmatrix}\right) + 1}\right) \begin{bmatrix} x_k \\ -1 \end{bmatrix}$$

$$\nabla f(s,r) = \frac{1}{K}\sum_{k=1}^{K}\left(1 - y_k - \frac{1}{\exp\left(\begin{bmatrix} x_k \\ -1 \end{bmatrix}^T \begin{bmatrix} s \\ r \end{bmatrix}\right) + 1}\right) \begin{bmatrix} x_k \\ -1 \end{bmatrix}$$

Figura 13: Decomposição da função objetivo para obtenção do gradiente.

Resultados do algoritmo *Gradient descent* para o *dataset* 1

$s = (1.3495, 1.0540)$ and $r = 4.8815$

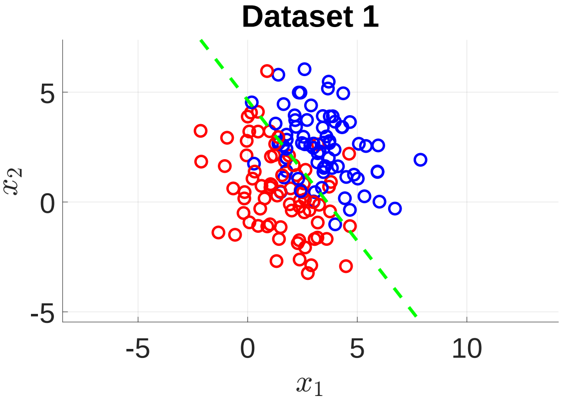


Figura 14: *Dataset* 1 e a correspondente reta $\{x \in \mathrm{R}^2 : s^T x = r\}$.

Resultados do algoritmo *Gradient descent* para o *dataset* 1

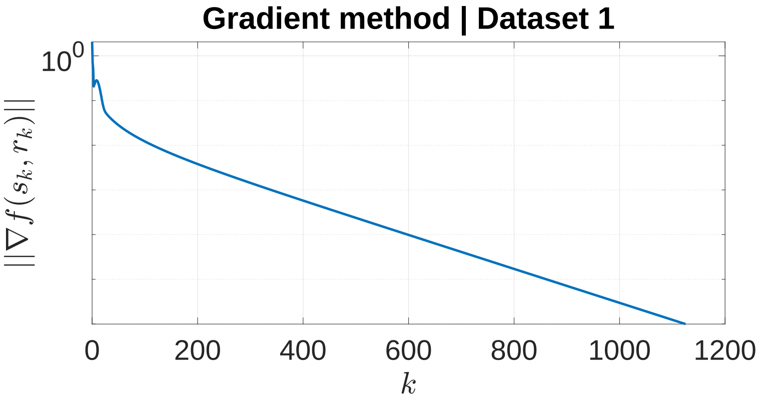


Figura 15: Norma do gradiente ao longo das iterações para o *dataset* 1.

Resultados do algoritmo *Gradient descent* para o *dataset* 2

$s = (0.7402, 2.3577)$ and $r = 4.5553$

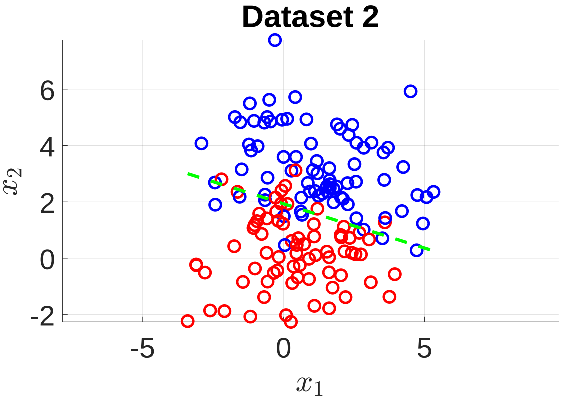


Figura 16: *Dataset* 2 e a correspondente reta $\{x \in \mathrm{R}^2 : s^T x = r\}$.

Resultados do algoritmo *Gradient descent* para o *dataset* 2

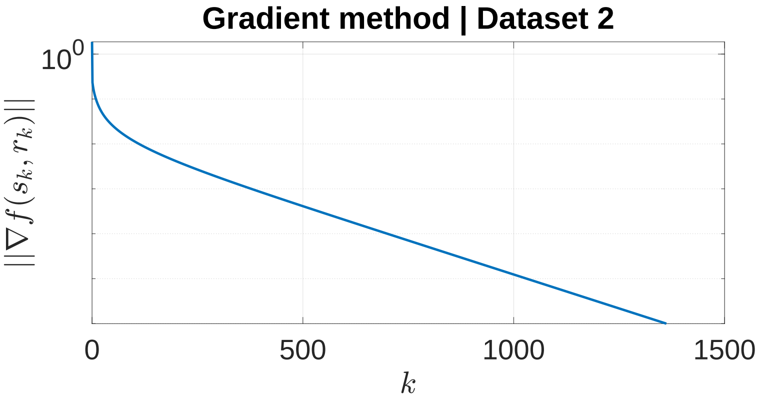


Figura 17: Norma do gradiente ao longo das iterações para o *dataset* 2.

Resultados do algoritmo *Gradient descent* para o *dataset* 3

$r = 4.7984$

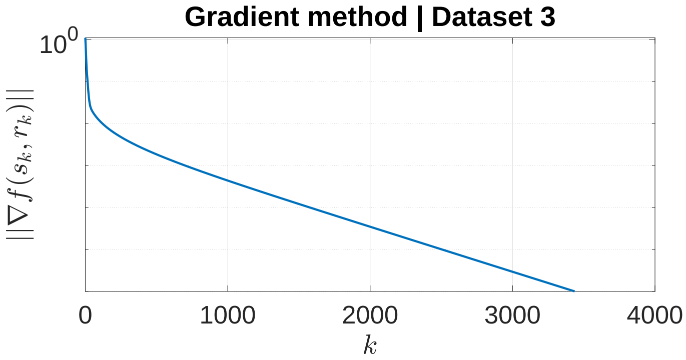


Figura 18: Norma do gradiente ao longo das iterações para o *dataset* 3.

Resultados do algoritmo *Gradient descent* para o *dataset* 4

$r = 7.6701$

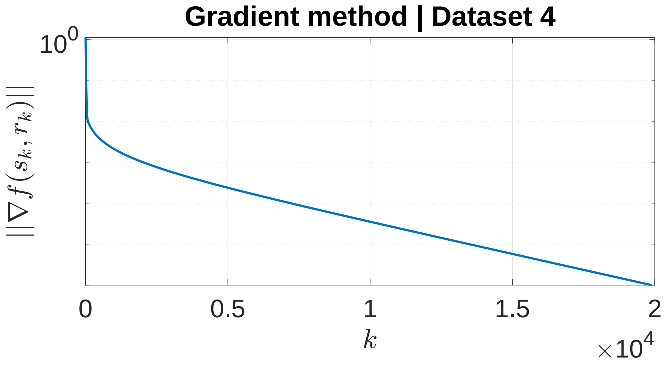


Figura 19: Norma do gradiente ao longo das iterações para o *dataset* 4.

Matriz Hessiana da função objetivo

$$f(s,r) = \frac{1}{K}\sum_{k=1}^{K}\left(\log\left(1+\exp\left(s^T x_k - r\right)\right) - y_k\left(s^T x_k - r\right)\right)$$

pode ser escrita como

$$f(x) = \sum_{k=1}^{K}\left(\phi\left(a_k^T x\right) + \frac{1}{K}\begin{bmatrix} -y_k x_k \\ y_k \end{bmatrix}^T x\right)$$

onde

$$\phi(z) = \frac{1}{K}\log(1+\exp(z))$$

$$a_k = [x_k \quad -1]^T$$

$$x = [s \quad r]^T$$

Matriz Hessiana da função objetivo

Da Hessiana de uma função afim e da Hessiana da soma,

$$\nabla^2 f(x) = \nabla^2 \left(\sum_{k=1}^{K} \phi \left(a_k^T x \right) \right).$$

Pela regra da derivada da composta,

$$\nabla f(x) = \sum_{k=1}^{K} \phi \left(a_k^T x \right) a_k.$$

Matriz Hessiana da função objetivo

Derivando novamente, chega-se a

$$\nabla^2 f(x) = \sum_{k=1}^{K} a_k \ddot{\phi}(a_k^T x) a_k^T,$$

pelo que

$$\nabla^2 f(x) = \begin{bmatrix} a_1 & a_2 & \dots & a_K \end{bmatrix} \begin{bmatrix} \ddot{\phi}(a_1^T x) & & & \\ & \ddot{\phi}(a_2^T x) & & \\ & & \dots & \\ & & & \ddot{\phi}(a_K^T x) \end{bmatrix} \begin{bmatrix} a_1^T \\ a_2^T \\ \dots \\ a_K^T \end{bmatrix}$$

com

$$\ddot{\phi}(z) = \frac{\exp(z)}{K\left[1 + \exp(z)\right]^2}.$$

Resultados do algoritmo de Newton

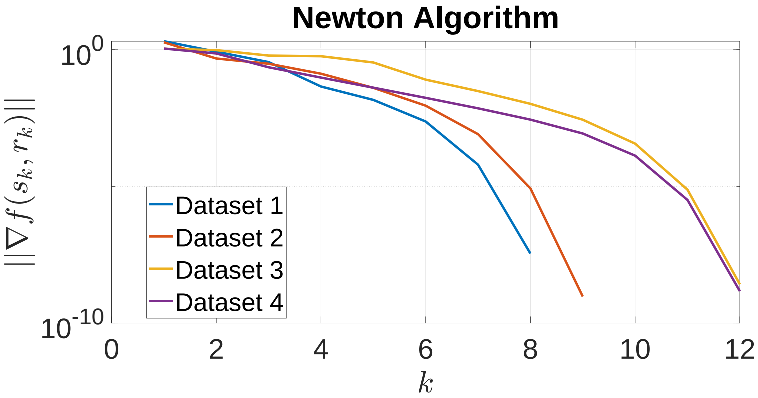


Figura 20: Norma do gradiente ao longo das iterações para todos os *datasets*.

Resultados do algoritmo de Newton

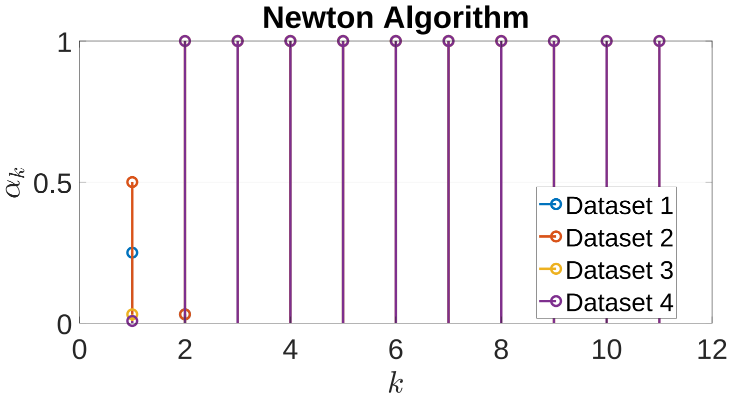


Figura 21: Valor do *stepsize* ao longo das iterações para todos os *datasets*.

Comparação dos resultados dos algoritmos *Gradient descent* e de Newton

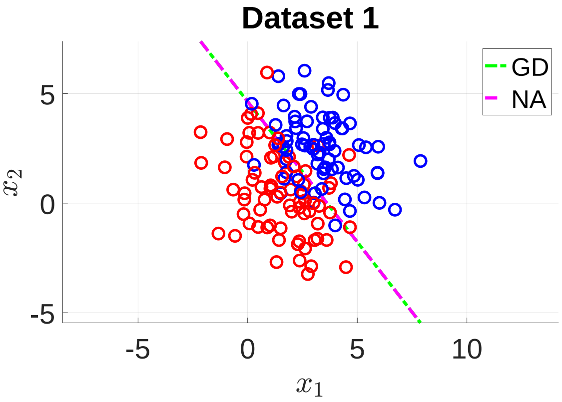


Figura 22: Comparação dos resultados obtidos para po *dataset* com ambos os métodos.

Comparação dos resultados dos algoritmos *Gradient descent* e de Newton

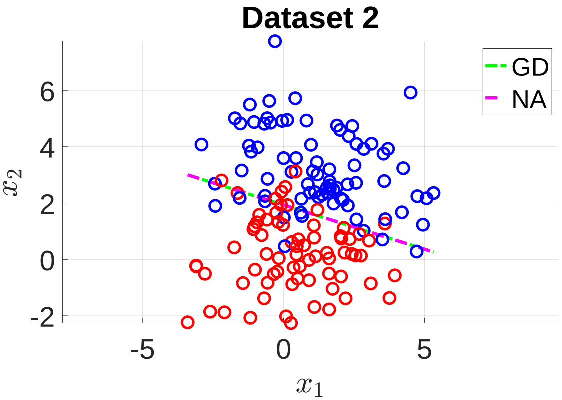


Figura 23: Comparação dos resultados obtidos para po *dataset* 2 com ambos os métodos.

Comparação dos resultados dos algoritmos *Gradient descent* e de Newton

Tabela 2: Dados de execução para cada método e *dataset*.

Método	*Dataset*	Iterações	Tempo [s]	Méd. tempo/iteração [s]
GD	1	1125	0.078	6.90×10^{-5}
NA	1	7	0.041	2.30×10^{-3}
GD	2	1362	0.082	6.04×10^{-5}
NA	2	8	0.014	3.16×10^{-3}
GD	3	3436	1.00	29.1×10^{-5}
NA	3	11	0.051	465×10^{-3}
GD	4	19892	169	851×10^{-5}
NA	4	11	8.51	774×10^{-3}

Conteúdo

Parte 3 - Introdução

Nesta parte pretende-se resolver o seguinte problema

$$\underset{\mathrm{y} \in \mathrm{R}^{Nk}}{\text{minimize}} \quad f(\mathrm{y}),$$

onde

$$f(\mathrm{y}) := \sum_{m=1}^{N} \sum_{n=m+1}^{N} (||\mathrm{y_m} - \mathrm{y_n}||_2 - D_{mn})^2 = \sum_{m=1}^{N} \sum_{n=m+1}^{N} f_{mn}(\mathrm{y})^2 \,,$$

e

$$D_{mn} = ||\mathrm{x_m} - \mathrm{x_n}||_2 \,.$$

- Não é um problema de otimização convexo!
- Será resolvido usando o método Levenberg-Marquardt (LM)

Part 3 - Task 2

Definindo $\mathrm{y_{m-n}} := \mathrm{y_m} - \mathrm{y_n}$ podemos escrever

$$f_{mn}(\mathrm{y}) := ||\mathrm{y_{m-n}}|| - D_{mn}$$

e

$$D_\mathrm{y} f_{mn}(\mathrm{y}) = \frac{\left[0_{1\times(m-1)k} \ \mathrm{y_{m-n}}^T \ 0_{1\times(n-m-1)k} \ -\mathrm{y_{m-n}}^T \ 0_{1\times(N-n)k}\right]}{||\mathrm{y_{m-n}}||} .$$

- Tanto $f_{mn}(\mathrm{y})$ como $D_\mathrm{y} f_{mn}(\mathrm{y})$ são escritos em função de cada $\mathrm{y_{m-n}}$ e da sua norma.

Part 3 - Task 2

A função de otimização é

$$f(\mathrm{y}) = \sum_{m=1}^{N} \sum_{n=m+1}^{N} f_{mn}(\mathrm{y})^2 ,$$

e o seu jacobiano é dado por

$$\begin{aligned} D_{\mathrm{y}} f(\mathrm{y}) &= \sum_{m=1}^{N} \sum_{n=m+1}^{N} D_u(u^2)\Big|_{u=f_{mn}(\mathrm{y})} D_{\mathrm{y}} f_{mn}(\mathrm{y}) \\ &= \sum_{m=1}^{N} \sum_{n=m+1}^{N} 2 f_{mn}(\mathrm{y}) D_{\mathrm{y}} f_{mn}(\mathrm{y}) . \end{aligned}$$

- Tanto $f(\mathrm{y})$ como $D_{\mathrm{y}} f(\mathrm{y})$ são escritos em função de cada $f_{mn}(\mathrm{y})$ e $D_{\mathrm{y}} f_{mn}(\mathrm{y})$.

Part 3 - Task 2

Para cada iteração do método LM é necessário calcular A e b dados por

$$\mathrm{A} := \begin{bmatrix} D_{\mathrm{y}} f_{1,1}(\mathrm{y}) \\ D_{\mathrm{y}} f_{1,2}(\mathrm{y}) \\ \vdots \\ D_{\mathrm{y}} f_{N-1,N}(\mathrm{y}) \\ \sqrt{\lambda}\mathrm{I}_{Nk\times Nk} \end{bmatrix} \quad \text{e} \quad \mathrm{b} := \begin{bmatrix} D_{\mathrm{y}} f_{1,1}(\mathrm{y})\mathrm{y} - f_{1,1}(\mathrm{y}) \\ D_{\mathrm{y}} f_{1,2}(\mathrm{y})\mathrm{y} - f_{1,2}(\mathrm{y}) \\ \vdots \\ D_{\mathrm{y}} f_{N-1,N}(\mathrm{y})\mathrm{y} - f_{N-1,N}(\mathrm{y}) \\ \sqrt{\lambda}\mathrm{y} \end{bmatrix} .$$

Note-se que

$$D_{\mathrm{y}} f_{mn}(\mathrm{y})\mathrm{y} - f_{mn}(\mathrm{y}) = \mathrm{y}_{\mathrm{m-n}}(\mathrm{y}_{\mathrm{m}} - \mathrm{y}_{\mathrm{n}}) - ||\mathrm{y}_{\mathrm{m-n}}|| + D_{mn} = D_{mn} ,$$

logo

$$\mathrm{b} = \begin{bmatrix} D_{1,1} & D_{1,2} & \dots & D_{N-1,N} & \sqrt{\lambda}\mathrm{y} \end{bmatrix} .$$

- Fração considerável das entradas de b é constante, pelo que pode ser calculada uma única vez.

Part 3 - Task 2

Em cada iteração do método LM é necessário calcular f(y),$||D_\mathrm{y} f(\mathrm{y})||$, A, e b $\rightarrow$ desenvolvida a função:

$$[f(\mathrm{y}), ||D_\mathrm{y} f(\mathrm{y})||, \mathrm{A}, \mathrm{b}] = \mathit{objectiveF(y)}\,.$$

- Calculadas apenas parte de A e b independentes de λ;
- $f(\mathrm{y}), ||D_\mathrm{y} f(\mathrm{y})||, \mathrm{A}$, e b calculados simultaneamente;
- As quantidades f(y),$||D_\mathrm{y} f(\mathrm{y})||$, e A são calculadas iterativamente para todos os pares

  $$(m, n) : m \in \{1, \ldots, N-1\}, n \in \{m+1, \ldots, N\}\,,$$

  prefazendo um total de $N(N/2 - 1)$ iterações;
- b é calculado apenas na primeira instância desta função.

Part 3 - Task 2

Para a i-th iteração (m, n):

- $\mathrm{y_{m-n}} \leftarrow \mathrm{y_m} - \mathrm{y_n}$
- $||\mathrm{y_{m-n}}|| \leftarrow \sqrt{\mathrm{y_{m-n}}^T \mathrm{y_{m-n}}}$
- $f_{mn}(\mathrm{y}) \leftarrow ||\mathrm{y_{m-n}}|| - \mathrm{b}(i)$
- $D_\mathrm{y} f_{mn}(\mathrm{y}) \leftarrow \frac{[0_{1\times(m-1)k} \ \mathrm{y_{m-n}}^T \ 0_{1\times(n-m-1)k} \ -\mathrm{y_{m-n}}^T \ 0_{1\times(N-n)k}]}{||\mathrm{y_{m-n}}||}$
- $f(\mathrm{y}) \leftarrow f(\mathrm{y}) + f_{mn}(\mathrm{y})^2$
- $D_\mathrm{y} f(\mathrm{y}) \leftarrow D_\mathrm{y} f(\mathrm{y}) + 2 f_{mn}(\mathrm{y}) D_\mathrm{y} f_{mn}(\mathrm{y})$
- $\mathrm{row}_i(\mathrm{A}) \leftarrow D_\mathrm{y} f_{mn}(\mathrm{y})$

Melhoramento de 2 ordens de grandeza no esforço computacional em relação a uma primeira implementação ingénua.

Part 3 - Task 4

Não é dada inicialização:

- Resolver o problema com LM para vários y_0 aleatórios
- Cada LM pode ser calculado em paralelo
- Escolhe-se a melhor solução

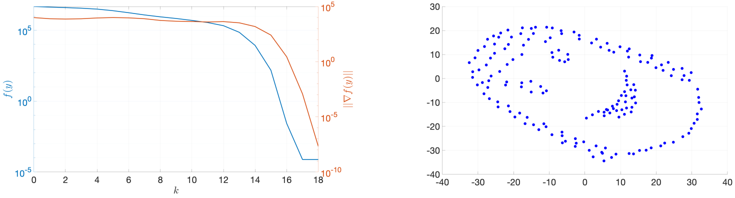


Figura 24: Melhor solução em 24 inicializações distintas para $k = 2$.

Part 3 - Task 4

Verificou-se que:

- Para todas as 24 inicializações obtém-se o mesmo valor de f
- As soluções verificam $f(\mathrm{y}_{sol}) = 7.6430 \times 10^{-5}$
- A função de otimização é não-negativa
- Se se reduzir ϵ, $f(\mathrm{y}_{sol})$ aproxima-se de 0
- Na prática, pode ser considerado um mínimo global ($f(\mathrm{y}_{sol}) << D_{ij}$)

Part 3 - Task 4 - Unicidade da solução

Seja

$$\bar{\mathrm{y}} = \mathrm{col}(\mathcal{T}\mathrm{y}_1 + \mathrm{w}, \dots, \mathcal{T}\mathrm{y}_\mathrm{N} + \mathrm{w}) \,,$$

onde $\mathcal{T} \in \mathbb{R}^{k \times k}$ é uma matriz de rotação e $\mathrm{w} \in \mathbb{R}^k$. É evidente que

$$||\bar{\mathrm{y}}_\mathrm{m} - \bar{\mathrm{y}}_\mathrm{n}|| = ||\mathrm{y}_\mathrm{m} - \mathrm{y}_\mathrm{n}||$$

para qualquer par $(m, n) : m \in \{1, \dots, N\}, n \in \{1, \dots, N\}$. Logo

$$f(\bar{\mathrm{y}}) = f(\mathrm{y}) \,.$$

- Se for encontrada uma solução, então existe uma infinidade de soluções.
- Logo, a solução encontrada não é única.

Part 3 - Task 4 - Unicidade da solução

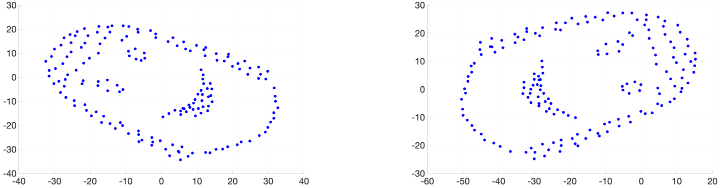


Figura 25: Duas melhores soluções em 24 inicializações distintas para $k = 2$.

- Uma solução pode ser obtida da outra por via de uma rotação e translação de cada $\mathrm{y_i}$, $i \in \{1, \ldots, N\}$.

Part 3 - Task 4 - Unicidade da solução

- Será que todas as soluções são da forma

$$\bar{\mathrm{y}} = \operatorname{col}(\mathcal{T}\mathrm{y}_1 + \mathrm{w}, \ldots, \mathcal{T}\mathrm{y}_\mathrm{N} + \mathrm{w})\ ?$$

- Não! A reflexão de cada y_i, $i \in \{1, \ldots, N\}$ em relação ao eixo da primeira coordenada origina um valor igual da função de custo.

Part 3 - Task 4 - cMDS

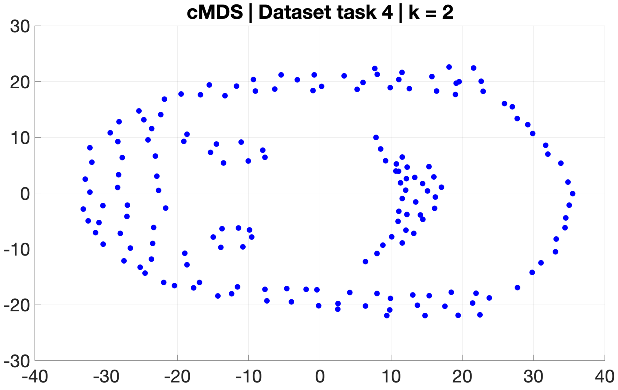


Figura 26: Solução método cMDS.

- A solução corresponde a uma translação e rotação das duas soluções anteriores.