Otimização Sem Restrições

M. Fernanda P. Costa

Departamento de Matemática
Universidade do Minho

Outline

Otimização

- área da matemática que se dedica ao estudo de problemas de minimização ou maximização de uma função de várias variáveis, dentro de um dado conjunto admissível;
- os problemas de otimização surgem quando se pretende calcular a melhor solução, dentro de um conjunto de soluções alternativas, para um problema.

Os **problemas de otimização** surgem em diversas áreas:

- ciências, engenharias, finanças, medicina, economia,
- big-data: machine learning ... (problemas de classificação, reconhecimento de padrões, reconhecimento de imagens, ...)

As componentes de um **problema de otimização** são:

- **variáveis de decisão (ou parâmetros)**, $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, do problema
- **restrições** - definem valores aceitáveis para as variáveis w;
- **função objetivo** $F(w)$ - mede a qualidade das potenciais soluções.

Tipos de problemas de otimização

Os problemas de otimização podem ser classificados em dois grandes grupos:

- problemas de otimização sem restrições;
- problemas de otimização com restrições.

Problema de otimização sem restrições:

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}} \; F(w) \qquad (P_{SR})$$

- $w = (w_1, w_2, \ldots, w_d)$ são as variáveis de decisão
- $F : \mathbb{R}^d \to \mathbb{R}$ é a função objetivo (medida de desempenho)

Problema de otimização com restrições (na sua forma mais geral):

$$\begin{aligned} \underset{w\in\mathbb{R}^d}{\text{minimizar}} \quad & F(w) \\ \text{sujeito a} \quad & c_n(w) = 0, \quad n \in \mathcal{E} = \{1,\ldots,j\} \\ & c_n(w) \geq 0, \quad n \in \mathcal{I} = \{j+1,\ldots,N\} \end{aligned} \tag{P_{CR}}$$

- $w = (w_1, w_2, \ldots, w_d)$ são as variáveis de decisão
- $F : \mathbb{R}^d \to \mathbb{R}$ é a função objetivo (medida de desempenho)
  (loss or cost function in ML)
- $c_n : \mathbb{R}^d \to \mathbb{R}$ com $n \in \mathcal{E}$, são as funções de restrição de igualdade
- $c_n : \mathbb{R}^d \to \mathbb{R}$ com $n \in \mathcal{I}$, são as funções de restrição de desigualdade

Definição: Chama-se **ponto admissível** para (P_{CR}) a um ponto que verifica todas as restrições.

Definição: Ao conjunto de todos os pontos admissíveis para (P_{CR}), chama-se **conjunto admissível** e será denotado por $\mathcal{D}$.

$$\mathcal{D} = \{w \in \mathbb{R}^d : c_n(w) = 0, n \in \mathcal{E}; c_n(w) \geq 0, n \in \mathcal{I}\}$$

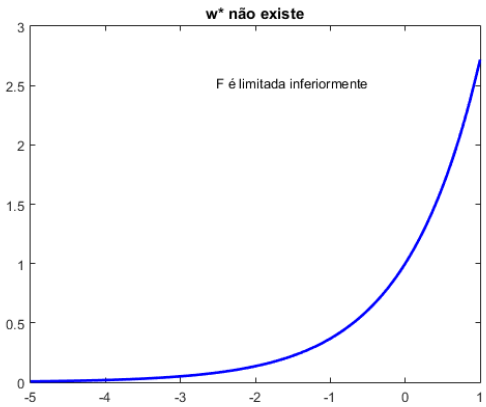
solução ótima e valor ótimo

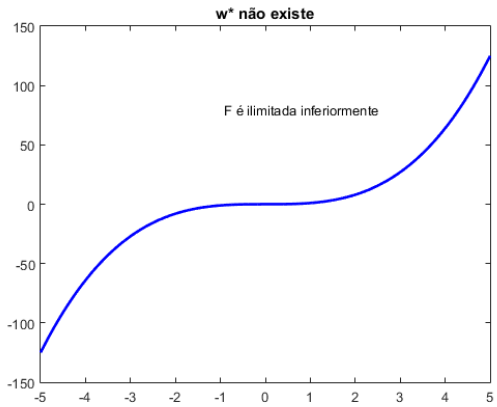
- $w^* \in \mathbb{R}^d$ é designado por solução ótima (ou a "solução", a "solução global", ou "argmin F sobre $\mathcal{D}$") se é um ponto em $\mathcal{D}$ que satisfaz a condição:

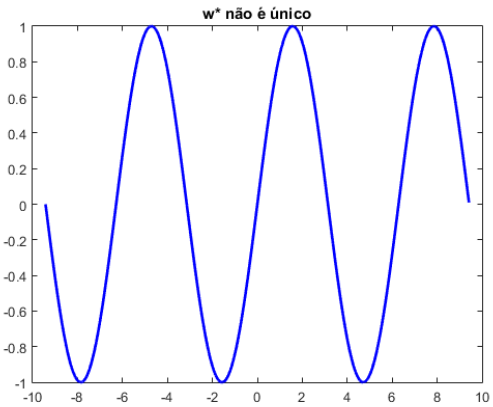
$$F(w^*) \leq F(w), \quad \text{para todo } w \in \mathcal{D}$$

e $F(w^*)$ é designado por valor ótimo do problema.

▷ Há problemas em que a solução ótima pode não existir ou não ser única.

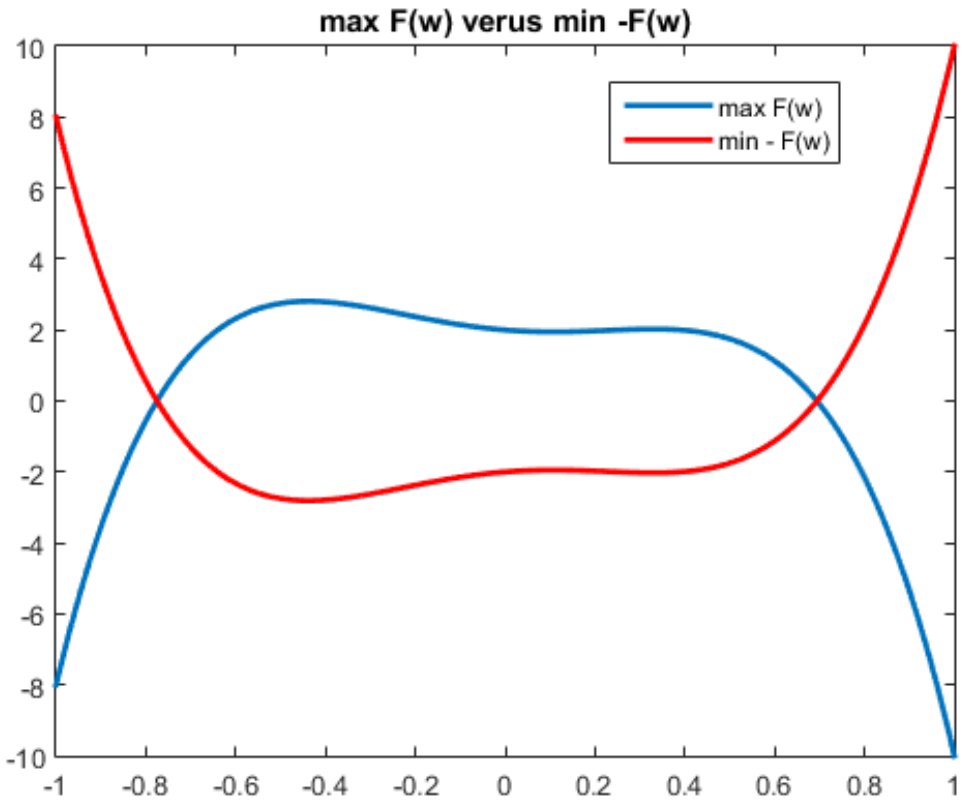






maximização verus minimização

- Qualquer problema de *maximização* pode ser reformulado como um problema de minimização: maximizar $F(w) = -$minimizar $-F(w)$.



$\triangleright$ o ponto w^* onde F atinge o seu máximo $F(w^*)$ é o mesmo onde $-F$ atinge o seu mínimo $-F(w^*)$.

minimizante local e global

Considere o problema de otimização (com ou sem restrições), e $w^* \in \mathcal{D}$.

- w^* é um minimizante global se

$$F(w^*) \leq F(w), \quad \forall w \in \mathcal{D};$$

- w^* é um minimizante global estrito se

$$F(w^*) < F(w), \quad \forall w \in \mathcal{D},\ w \neq w^*;$$

- w^* é um minimizante local se existir $\epsilon > 0$:

$$F(w^*) \leq F(w), \ \forall w \in B(w^*, \varepsilon) \cap \mathcal{D}$$

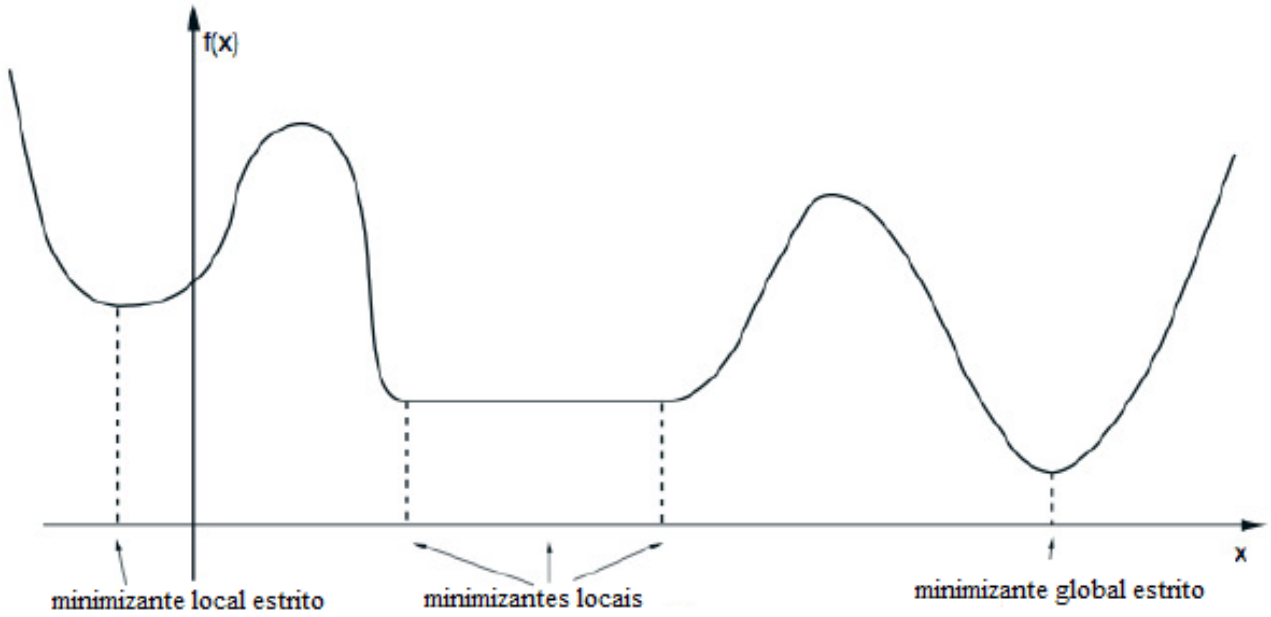
- w^* é um minimizante local estrito se existir $\epsilon > 0$:

$$F(w^*) < F(w), \quad \forall w \in B(w^*, \varepsilon) \cap \mathcal{D},\ w \neq w^*;$$

▷ Todo o minimizante global é um minimizante local.

nota: $B(w^*; \epsilon) := \{w \in \mathbb{R}^d : \|w^* - w\| \leq \epsilon\}$, vizinhança de w^* de raio ϵ.

Exemplo:



obs: Não confundir minimizante com mínimo. Se w^* é um minimizante de F então $F(w^*)$ é o respetivo mínimo: o minimizante é um vetor em $\mathbb{R}^d$ que minimiza a função F; o mínimo será $F(w^*)$ (um valor real).

Problema de otimização sem restrições

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}}\ F(w)$$

- Uma classe especial deste tipo de problemas é a dos **problemas de mínimos quadrados**:

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}}\ F(w) := \sum_{n=1}^{N} (f^n(w))^2$$

- No contexto do "Ajuste de Dados", ou seja, "Regressão", cada função $f^n(w)$ define um resíduo.

Exemplos de aplicações de otimização sem restrições:

- **Regressão:** dados N pares de pontos (x^n, y^n), o vetor dos parâmetros $w \in \mathbb{R}^d$ que melhor ajustam a função (modelo matemático) $\phi(w; x)$ aos dados, é dado por:

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}}\, F(w) := \sum_{n=1}^{N} \underbrace{(\phi(w; x^n) - y^n)}_{f^n(w)}{}^2$$

onde

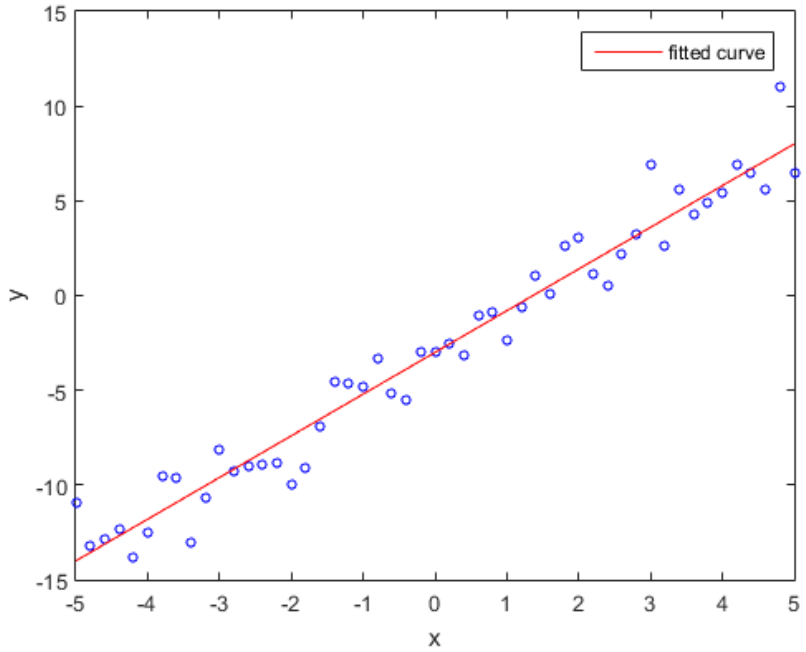
- $\widetilde{y}^n = \Phi(w; x^n)$ é o valor previsto pelo modelo;
- y^n - é o valor conhecido associado a x^n ;

▷ O problema é de mínimos quadrados linear se a função ϕ é linear nas componentes de w:

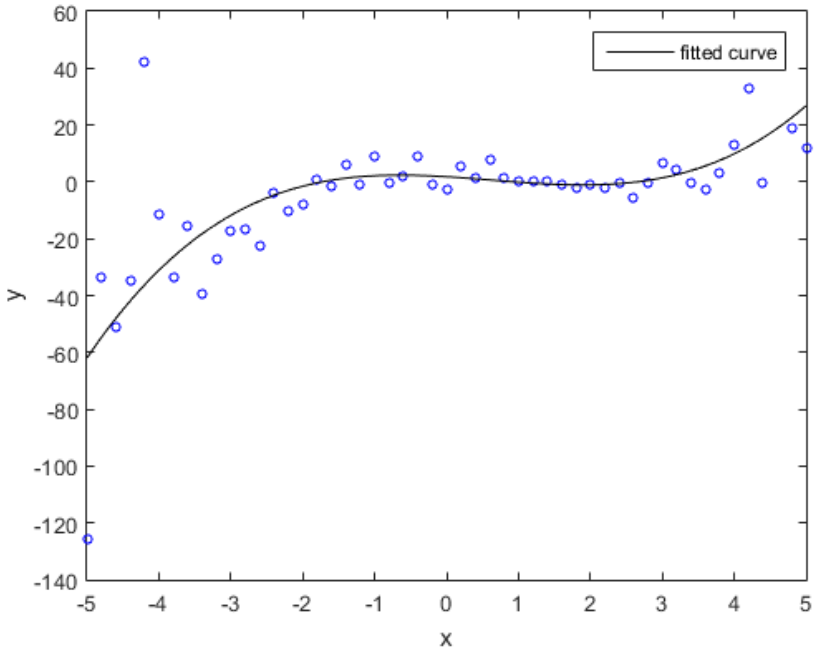
$$\phi(w; x^n) = w_1 g_1(x^n) + w_2 g_2(x^n) + \cdots + w_d g_d(x^n)$$

onde cada função g_i depende apenas de x^n.

- Exemplos de ajuste:



$$\phi(w;x) = w_1 + w_2 x$$
$$f^n(w) = \phi(w;x^n) - y^n$$



$$\phi(w;x) = w_1 + w_2 x + w_3 x^2 + w_4 x^3$$
$$f^n(w) = \phi(w;x^n) - y^n$$

Derivadas da função $F : \mathbb{R}^d \to \mathbb{R}$

- Vetor gradiente (1ª derivada) de F:

$$\nabla F(w) = \begin{bmatrix} \frac{\partial F}{\partial w_1} \\ \vdots \\ \frac{\partial F}{\partial w_d} \end{bmatrix} \text{ vetor de } \mathbb{R}^d$$

- Matriz hessiana (2ª derivada) de F:

$$\nabla^2 F(w) = \begin{bmatrix} \frac{\partial^2 F}{\partial w_1^2} & \cdots & \frac{\partial^2 F}{\partial w_d \partial w_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial w_1 \partial w_d} & \cdots & \frac{\partial^2 F}{\partial w_d^2} \end{bmatrix} \text{ matriz simétrica } d \times d$$

Exemplo: Determinar o vetor gradiente e a matriz hessiana das seguintes funções $F : \mathbb{R}^d \to \mathbb{R}$:

- Função Linear: $F(w) = w^T q$, onde $q \in \mathbb{R}^d$, $q \neq 0$, é um vetor constante

  Função Afim: $F(w) = w^T q + p$, onde $q \in \mathbb{R}^d$, $q \neq 0$, $p \in \mathbb{R}$

  Resolução:

  $$\nabla F(w) = q, \quad \nabla^2 F(w) = \mathbf{0}$$

  Obs: $\mathbf{0}$ denota a matriz nula $d \times d$

- Função Quadrática: $F(w) = \frac{1}{2} w^T Q w + w^T q + p$, onde Q é uma matriz simétrica $d \times d$

  Resolução:

  $$\nabla F(w) = Qw + q, \quad \nabla^2 F(w) = Q$$

- Determinar o vetor gradiente e a matriz hessiana da função $F : \mathbb{R}^2 \to \mathbb{R}$ dada por $F(w) = w_1^4 + w_1 w_2 + (1 + w_2)^2$

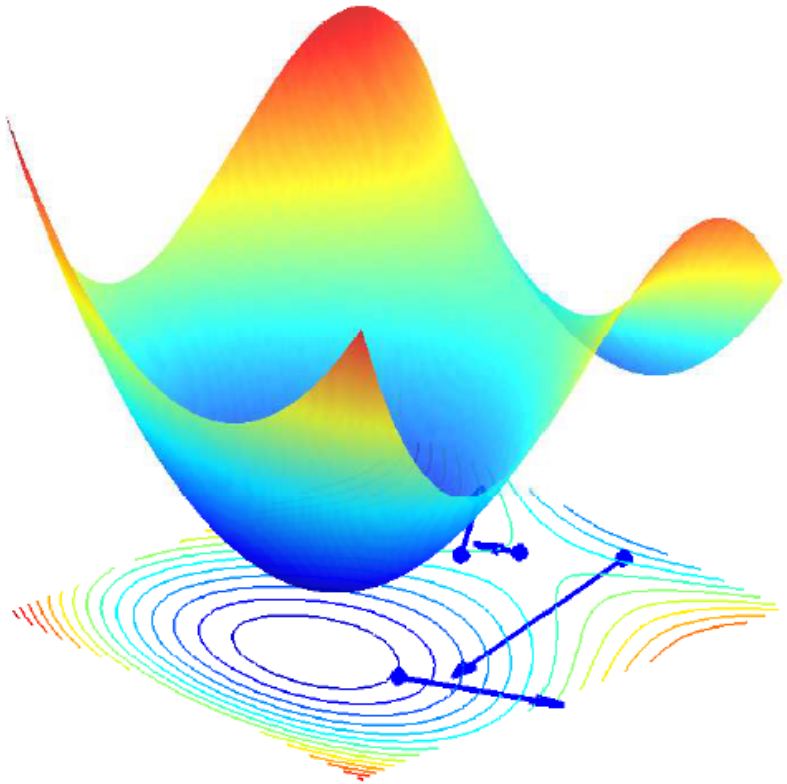
Resolução:

$$\nabla F(w) = \begin{bmatrix} \frac{\partial F}{\partial w_1} \\ \frac{\partial F}{\partial w_2} \end{bmatrix} = \begin{bmatrix} 4w_1^3 + w_2 \\ w_1 + 2(1 + w_2) \end{bmatrix}$$

$$\nabla^2 F(w) = \begin{bmatrix} \frac{\partial^2 F}{\partial w_1^2} & \frac{\partial^2 F}{\partial w_2 w_1} \\ \frac{\partial^2 F}{\partial w_1 w_2} & \frac{\partial^2 F}{\partial w_2^2} \end{bmatrix} = \begin{bmatrix} 12w_1^2 & 1 \\ 1 & 2 \end{bmatrix}$$

Propriedade do Vetor Gradiente

O vetor gradiente é perpendicular à curva de nível e aponta na direção de maior crescimento da função.



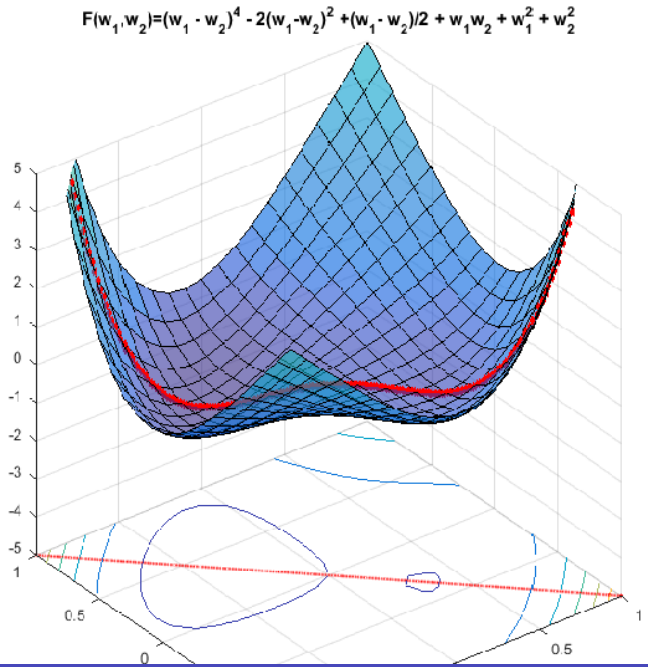
Restrição de $F : \mathbb{R}^d \to \mathbb{R}$ aos pontos de uma reta

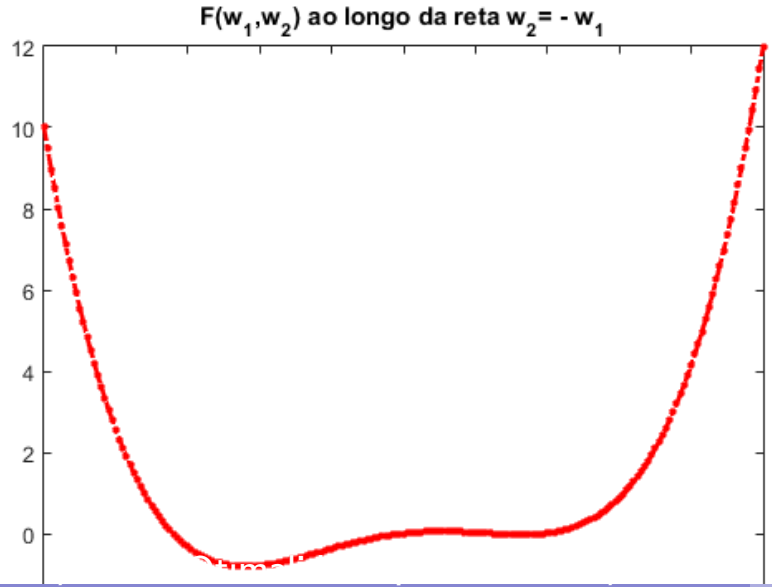
Dado o ponto $\overline{w} \in \mathbb{R}^d$ e o vetor $s \in \mathbb{R}^d$, a reta que passa em $\overline{w}$ e tem a direção de s é definida por:

$$w = \overline{w} + \eta\, s, \quad \eta \in \mathbb{R}.$$

A restrição de F aos pontos da reta é definida por:

$$g(\eta) := F(\overline{w} + \eta\, s)$$





$$\underset{w \in \mathbb{R}^d}{\text{minimizar}} \; F(w)$$

Usa a restrição de $F(w)$ ao longo de uma reta ...
Recordar condições suficientes para um mínimo local de uma função de 1 variável, $g(\eta) := F(\overline{w} + \eta\, s)$:

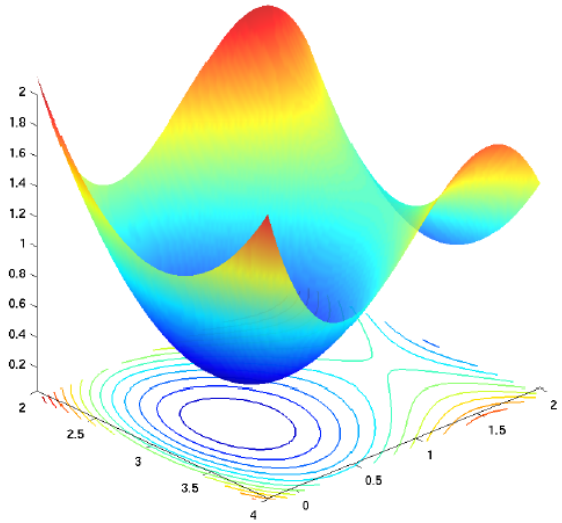
$$\frac{dg}{d\eta} = 0, \quad \text{e} \quad \frac{d^2 g}{d\eta^2} > 0$$

- Primeira derivada de g:
  $\frac{dg}{d\eta} = s^T \nabla F(\overline{w} + \eta\, s)$, declive de F ao longo da direção s
- Segunda derivada de g:
  $\frac{d^2 g}{d\eta^2} = s^T \nabla^2 F(\overline{w} + \eta\, s) s$, curvatura de F ao longo da direção s

Condições de otimalidade

Assume-se que F é continuamente diferenciável até à 2ª ordem.

Teorema (Condição necessária de 1ª ordem para minimizante)
Se w^* é um minimizante local de F então $\nabla F(w^*) = 0$.



Definição (Ponto estacionário)
Um ponto w^* que satisfaça a condição $\nabla F(w^*) = 0$ é designado por ponto estacionário de F.

Condições de otimalidade

Teorema (Condição necessária de 2ª ordem para minimizante)
Se w^* é um minimizante local de F então $\nabla F(w^*) = 0$ e $\nabla^2 F(w^*)$ é semi-definida positiva.

Teorema (Condições suficientes de 2ª ordem para minimizante)
Se $\nabla F(w^*) = 0$ e $\nabla^2 F(w^*)$ é definida positiva então w^* é um minimizante local de F.

Recordar: Seja $A \in \mathbb{R}^{d \times d}$ uma matriz simétrica.

- A é definida positiva sse os valores próprios de A são positivos;
- A é definida positiva sse os determinantes das submatrizes principais de A são positivos;
- A é definida positiva sse $s^T A s > 0$ para todo $s \in \mathbb{R}^d$, $s \neq 0$
- A é semi-definida positiva sse os valores próprios de A são não negativos;
- A é semi-definida positiva sse os determinantes das submatrizes principais de A são não negativos;
- A é semi-definida positiva sse $s^T A s \geq 0$ para todo $s \in \mathbb{R}^d$, $s \neq 0$.

Teorema. Se F é convexa, então qualquer minimizante local w^* é um minimizante global de F. Mais ainda, se F é diferenciável, então qualquer ponto estacionário é minimizante global de F.

Analogamente, as condições necessárias e suficientes de 2ª ordem para um **maximizante** são:

- w^* maximizante $\Rightarrow \nabla F(w^*) = 0$ e $\nabla^2 F(w^*)$ semi-definida negativa.
- $\nabla F(w^*) = 0$ e $\nabla^2 F(w^*)$ definida negativa $\Rightarrow w^*$ maximizante

Recordar: Seja $A \in \mathbb{R}^{d \times d}$ uma matriz simétrica.

- A é definida negativa sse os valores próprios de A são negativos;
- A é definida negativa sse os determinantes das submatrizes principais de A de ordem ímpar forem negativos, e os determinantes das submatrizes principais de A de ordem par forem positivos;
- A é definida negativa sse $s^T As < 0$ para todo $s \in \mathbb{R}^d$, $s \neq 0$
- A é semi-definida negativa sse os valores próprios de A são não positivos;
- A é semi-definida negativa sse os determinantes das submatrizes principais de A de ordem ímpar forem não positivos, e os determinantes das submatrizes principais de A de ordem par forem não negativos;
- A é semi-definida negativa sse $s^T As \leq 0$ para todo $s \in \mathbb{R}^d$, $s \neq 0$.

Teorema (Ponto sela)
Seja $\overline{w}$ um ponto estacionário de F. Se $\nabla^2 F(\overline{w})$ é indefinida, então $\overline{w}$ é ponto sela de F.

Recordar: Seja $A \in \mathbb{R}^{d \times d}$ uma matriz simétrica.

- A é indefinida se não for nem semi-definida positiva nem semi-definida negativa.
- A é indefinida sse A tem valores próprios positivos e negativos.

Conclusão

Classificação dos pontos estacionários:

- se $\nabla^2 F(w^*)$ definida positiva, então w^* é **minimizante local**;
- se $\nabla^2 F(w^*)$ definida negativa, então w^* é **maximizante local**;
- se $\nabla^2 F(w^*)$ indefinida, então w^* é **ponto sela**;
- $\nabla^2 F(w^*)$ semi-definida positiva, então w^* ou é **minimizante local** ou é **ponto sela**;
- $\nabla^2 F(w^*)$ semi-definida negativa, então w^* ou é **maximizante local** ou **ponto sela**.

Exemplo:

Considere a função $F : \mathbb{R}^2 \to \mathbb{R}$ dada por $F(w) = (w_1^2 - w_2)(2w_1^2 - w_2)$.
Verificar que $\overline{w} = (0,0)^T$ é ponto estacionário mas não é minimizante de F.

Resolução:

$\nabla F(w_1, w_1) = \begin{bmatrix} 8w_1^3 - 6w_1 w_2 \\ -3w_1^2 + 2w_2 \end{bmatrix}$. Como $\nabla F(0,0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\overline{w} = (0,0)^T$ é ponto estacionário.

$\nabla^2 F(w_1, w_2) = \begin{bmatrix} 24w_1^2 - 6w_2 & -6w_1 \\ -6w_1 & 2 \end{bmatrix}$. Como $\nabla^2 F(0,0) = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$

é semi-definida positiva, então $\overline{w} = (0,0)^T$ ou é minimizante local ou é ponto sela.

Notar que, a função é negativa para $w_1^2 < w_2 < 2w_1^2$. Por exemplo, para $w_2 = \frac{3}{2}w_1^2$ tem-se $F(w_1, \frac{3}{2}w_1^2) = -\frac{1}{4}w_1^4 < 0$, para $w_1 \neq 0$. Donde se conclui que, para valores de w_1 na vizinhança de 0, $F(w_1, \frac{3}{2}w_1^2) < F(0,0) = 0$, pelo que $\overline{w} = (0,0)^T$ não é um minimizante local de F e sim ponto sela de F.

Exemplo:

Classifique os pontos estacionários da função $F : \mathbb{R}^2 \to \mathbb{R}$ dada por:

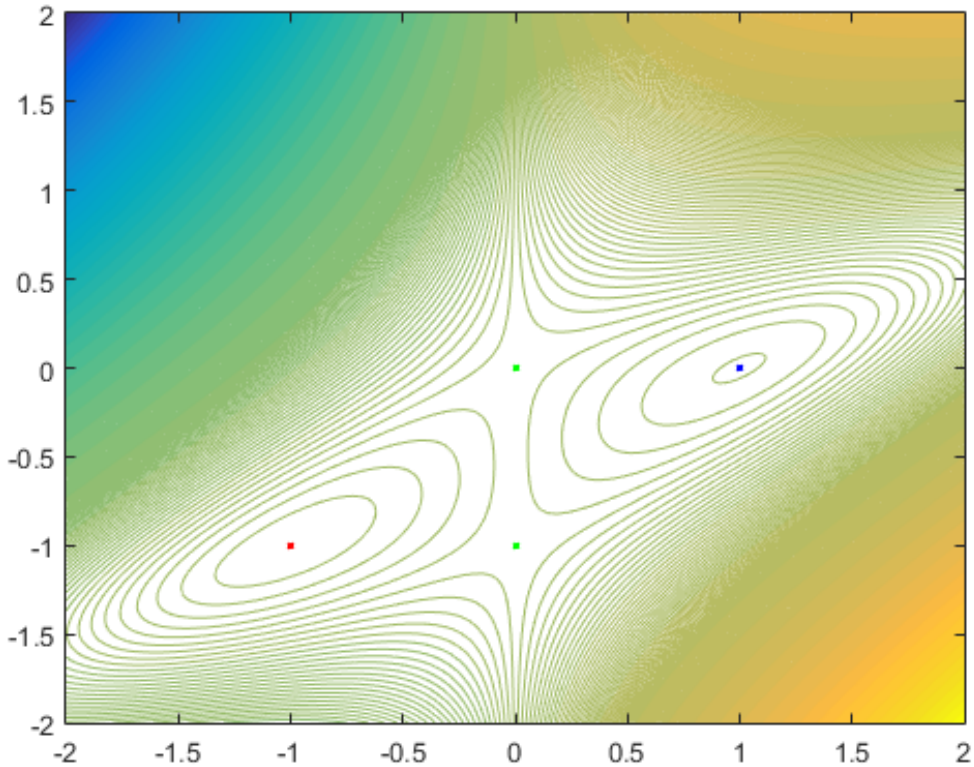
$$F(w_1, w_2) = 2w_1^3 - 3w_1^2 - 6w_1w_2(w_1 - w_2 - 1)$$

Resolução:

- **gradiente:** $\nabla F(w_1, w_2) = \begin{bmatrix} 6w_1^2 - 6w_1 - 12w_1w_2 + 6w_2^2 + 6w_2 \\ -6w_1^2 + 12w_1w_2 + 6w_1 \end{bmatrix}$
- **hessiana:**
  $\nabla^2 F(w_1, w_2) = \begin{bmatrix} 12w_1 - 12w_2 - 6 & -12w_1 + 12w_2 + 6 \\ -12w_1 + 12w_2 + 6 & 12w_1 \end{bmatrix}$
- **pontos estacionários:**
  $\nabla F(w_1, w_2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 6w_2^2 + 6w_2 \\ -6w_1^2 + 12w_1w_2 + 6w_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

  Os pontos estacionários são: $(0,0)^T$, $(0,-1)^T$, $(1,0)^T$, $(-1,-1)^T$

- $(0,0)^T$ é ponto sela; $(1,0)^T$ é minimizante local;
- $(0,-1)^T$ é ponto sela; $(-1,-1)^T$ é maximizante.



Exercício 1

1. Considere a função $F : \mathbb{R}^2 \to \mathbb{R}$ definida por

$$F(w_1, w_2) = \frac{1}{3}w_1^3 + \frac{1}{2}w_1^2 + 2w_1w_2 + \frac{1}{2}w_2^2 - w_2 + 9.$$

   Calcule os pontos estacionários da função, e verifique se são pontos mínimos locais ou pontos máximos locais. A função tem um ponto mínimo global?

2. Considere a função $F : \mathbb{R}^2 \to \mathbb{R}$ definida por

$$F(w_1, w_2) = 8w_1^2 + 3w_1w_2 + 7w_2^2 - 25w_1 + 31w_2 - 29.$$

   Calcule os pontos estacionários da função, e verifique se são pontos mínimos locais ou pontos máximos locais. A função tem um mínimo global?

Métodos iterativos para otimização sem restrições

Em geral, não se consegue resolver

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}} \; F(w)$$

analiticamente ... recorre-se a métodos iterativos.

Métodos iterativos para Otimização

- Começam a partir de uma aproximação inicial à solução, $w^{(1)}$
- Dado $w^{(k)}$, calculam um novo (melhor) ponto $w^{(k+1)}$, e este processo repete-se $(k = 1, 2, \dots)$
- Geram uma sucessão $\{w^{(k)}\}$ de aproximações na qual a função F decresce, que espera-se que convirja para a solução ótima w^*.

...convergência local versus global ...

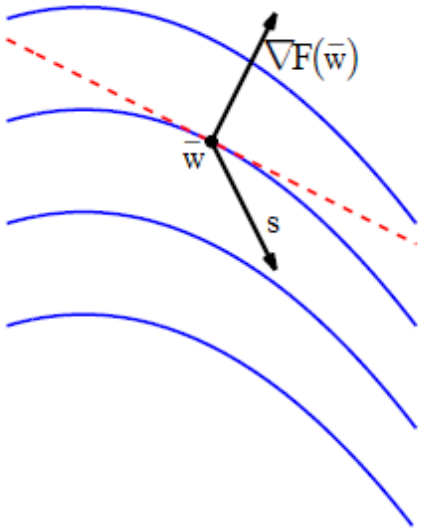
Diz-se que o método de otimização tem convergência de primeira ordem se a sucessão $\{w^{(k)}\}$ de aproximações à solução converge para um ponto estacionário de F.

- Usa-se o termo convergência global se o método tem convergência de primeira ordem qualquer que seja a aproximação inicial $w^{(1)}$ do processo iterativo.
- Usa-se o termo convergência local se o método tem convergência de primeira ordem apenas quando a aproximação inicial $w^{(1)}$ estiver suficientemente perto da solução.

Definição (Direção de descida)
Considere uma função $F : \mathbb{R}^d \to \mathbb{R}$ e um ponto $\overline{w} \in \mathbb{R}^d$. Uma direção $s \in \mathbb{R}^d \setminus \{0\}$ é uma direção de descida para F a partir de $\overline{w}$, se existe $\overline{\eta} > 0$ tal que

$$F(\overline{w} + \eta s) < F(\overline{w}), \text{ para todo } \eta \in (0, \overline{\eta})$$

Teorema Se $\nabla F(\overline{w})^T s < 0$, então s é uma direção de descida para F a partir de $\overline{w}$.



($\nabla F(\overline{w})^T s < 0 \Rightarrow$ o declive de F em $\overline{w}$ na direção de s é negativo)

Métodos de descida de procura unidirecional

Ideia:

- calcular uma direção de descida, $s^{(k)}$
- procurar uma redução no valor de F ao longo da direção $s^{(k)}$

Algoritmo: **Método de descida de procura undirecional geral**

1. Dar: $w^{(1)}$
2. Fazer $k = 1$
3. **Enquanto** ($w^{(k)}$ não satisfaz o critério de paragem)
4. Calcular uma direção de procura $s^{(k)}$ tal que $\nabla F(w^{(k)})^T s^{(k)} < 0$
5. Encontrar o comprimento do passo η_k tal que
$$F(w^{(k)} + \eta_k s^{(k)}) < F(w^{(k)})$$
6. Fazer $w^{(k+1)} = w^{(k)} + \eta_k s^{(k)}$
7. Fazer $k = k + 1$
8. **Fim enquanto**

Observações sobre o método de direções de descida geral:

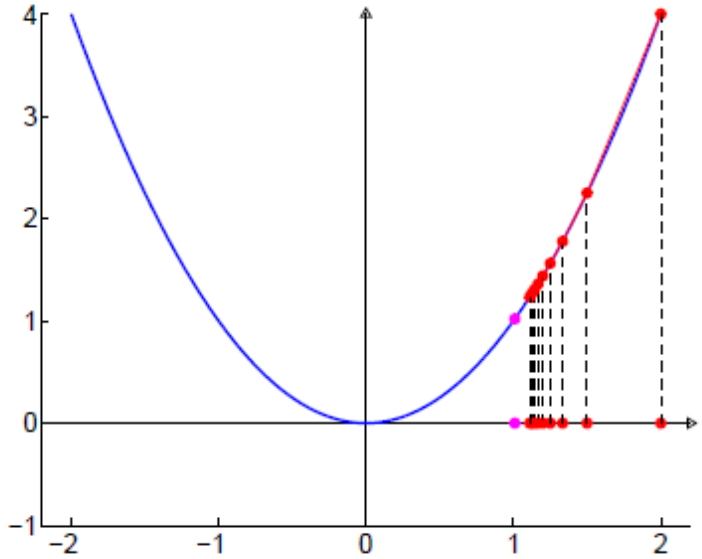
- existem várias escolhas possíveis para $s^{(k)}$;
- procura exata do η_k:

$$\underset{\eta \in \mathbb{R}}{\text{minimizar}} \; F(w^{(k)} + \eta s^{(k)})$$

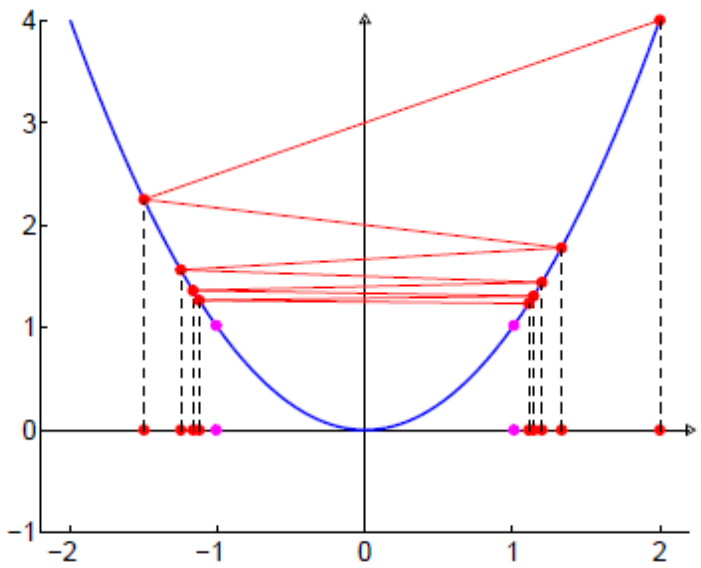
... é, em geral, impraticável, devido ao elevado custo computacional e, em geral, não é necessária. Na prática, recorre-se a técnicas de procura não exata.

- A condição de redução simples $F(w^{(k)} + \eta_k s^{(k)}) < F(w^{(k)})$, não garante convergência para um ponto estacionário (ver exemplo).

Exemplo: Considere $F : \mathbb{R} \to \mathbb{R}$ dada por $F(w) = w^2$ e as sucessões obtidas pelo algoritmo.



(a) passos muito curtos



(b) passos muito longos

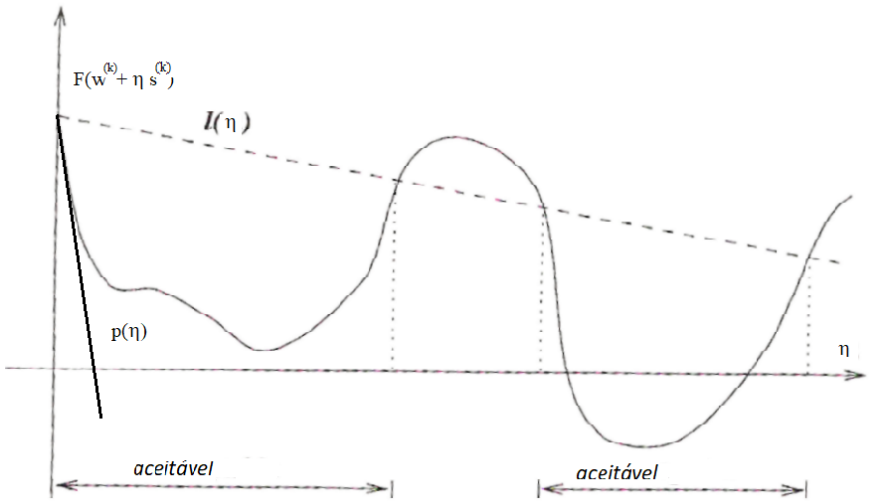
$w^{(k)} = 1 + \frac{1}{k+1}$
$w^{(k)} \to 1$
1 não é minimizante de F

$w^{(k)} = (-1)^k + \frac{(-1)^k}{k+1}$
$w^{(k)} \to 1$ (k par)
$w^{(k)} \to -1$ (k ímpar)
-1 e 1 não são minimizantes de F

Procura não exata - Condição de Armijo

Dados $w^{(k)}, s^{(k)} \in \mathbb{R}^d$ e $\delta \in (0,1)$. A **Condição de Armijo** encontra o η_k que origina uma redução significativa no valor de F, dada por:

$$F(w^{(k)} + \eta s^{(k)}) \leq \underbrace{F(w^{(k)}) + \delta \eta \nabla F^T \left(w^{(k)}\right) s^{(k)}}_{l(\eta)}$$



Condição de redução significativa

$\Rightarrow$ impede comprimentos de passos muito longos

Algoritmo - condição de Armijo com backtracking

Na prática, para prevenir que a condição Armijo seja verificada por comprimentos do passo muito pequenos (ver figura), aplica-se uma estratégia de bracktracking.

Algoritmo: condição de Armijo com backtracking

1. Dar $\overline{\eta} > 0$, $\delta \in (0, 1)$
2. Fazer $\eta \leftarrow \overline{\eta}$
3. **Enquanto** $F(w^{(k)} + \eta s^{(k)}) > F(w^{(k)}) + \delta \eta \nabla^T F\left(w^{(k)}\right) s^{(k)}$
4. Fazer $\eta \leftarrow \eta/2$
5. **Fim enquanto**
6. Fazer $\eta_k \leftarrow \eta$

Critérios de paragem

Critério 1 (proposto por Wolfe). Parar o algoritmo para otimização sem restrições se:

$$\underbrace{\left\|\nabla F(w^{(k)})\right\|}_{\text{medida de estacionaridade}} \leq \varepsilon_1$$

e

$$\underbrace{\frac{\left\|w^{(k)} - w^{(k-1)}\right\|}{\left\|w^{(k)}\right\|}}_{\text{erro relativo da aproximação}} \leq \varepsilon_2$$

e

$$\frac{\left|F(w^{(k)}) - F(w^{(k-1)})\right|}{\left|F(w^{(k)})\right|} \leq \varepsilon_3$$

$\varepsilon_1, \varepsilon_2, \varepsilon_3$ constantes positivas próximas de zero.

Critérios de paragem

Critério 2 (proposto por Gill e Murray). Parar o algoritmo para otimização sem restrições se:

$$\left\|\nabla F(w^{(k)})\right\| \leq \varepsilon^{\frac{1}{3}} \left(1 + \left|F(w^{(k)})\right|\right)$$

e

$$\left\|w^{(k)} - w^{(k-1)}\right\| \leq \varepsilon \left(1 + \left\|w^{(k)}\right\|\right)$$

e

$$\left|F(w^{(k)}) - F(w^{(k-1)})\right| \leq \varepsilon^2 \left(1 + \left|F(w^{(k)})\right|\right)$$

ε é uma constante positiva próxima de zero.

Observação: Este critério é de aplicação mais geral! Pode ser aplicado a problemas em que a solução ótima é o vetor nulo ou o valor ótimo da função objetivo é zero.

Método do gradiente

No **método do gradiente**, a direção de procura é dada pela direção de descida máxima:

$$s^{(k)} = -\nabla F(w^{(k)}).$$

Observar que:

- Entre todas as direções ao longo das quais F decresce, a direção oposta ao vetor gradiente é a de decrescimento mais acentuado. De facto, pela definição de produto interno entre dois vetores

$$s^{(k)\,T}\nabla F(w^{(k)}) = \|s^{(k)}\|\|\nabla F(w^{(k)})\|\cos(\theta),$$

onde θ é o ângulo entre os vetores $s^{(k)}$ e $\nabla F\left(w^{(k)}\right)$, e este termo atinge o seu valor mínimo quando $\cos(\theta) = -1$, ou seja, $\theta = \pi$, isto é, $s^{(k)} = -\nabla F\left(w^{(k)}\right)$.

Método do gradiente

Algoritmo: Método do Gradiente

1. Dar: $w^{(1)}$
2. Fazer $k = 1$
3. **Enquanto** ($w^{(k)}$ não satisfaz o critério de paragem)
4. Calcular a direção de descida máxima $s^{(k)} = -\nabla F(w^{(k)})$
5. Calcular o comprimento do passo η_k tal que

$$F(w^{(k)} + \eta_k s^{(k)}) < F(w^{(k)})$$

6. Fazer $w^{(k+1)} = w^{(k)} + \eta_k s^{(k)}$
7. Fazer $k = k + 1$
8. **Fim enquanto**

Propriedade do Método do gradiente

- Se no método do gradiente o comprimento do passo, η_k, é obtido por procura exata, então as sucessivas direções de descida máxima definem ângulos retos:

$$s^{(k+1)\,T} s^{(k)} = 0$$

e o método apresenta um comportamento em ziguezague que se traduz num processo muito lento quando já está perto do mínimo.

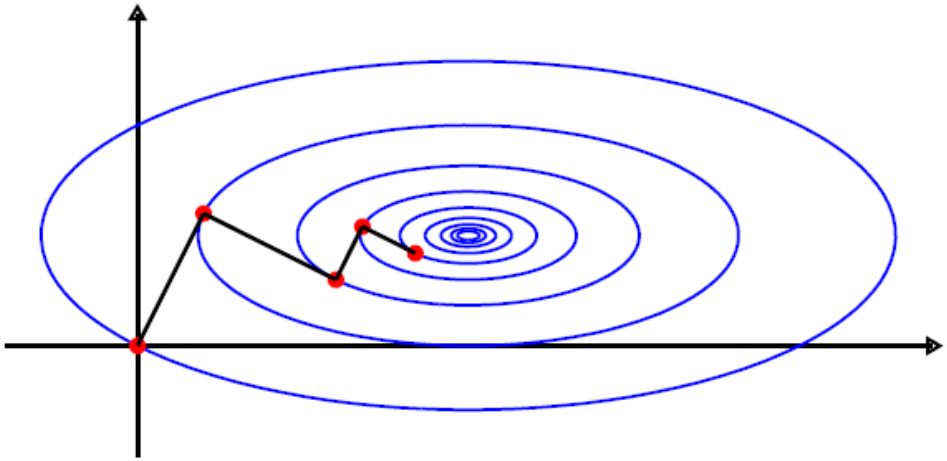


Figura. 4 iterações do algoritmo de direções de descida máxima com procura exata, ao minimizar uma função quadrática convexa

- Se no método do gradiente o comprimento do passo, η_k, é obtido por procura exata (ou por procura não exata:condição de Armijo) então o método é globalmente convergente.

Exercício 2

Resolva o problema

$$\underset{w \in \mathbb{R}^2}{\text{minimizar}} \; F(w) = w_1^2 + 2w_2^2$$

usando o método do gradiente. O processo iterativo deve ser iniciado com o ponto $(1, 1)$ e deve terminar quando o critério de paragem baseado na condição $\|\nabla F(w)\| \leq \varepsilon$ for verificado para $\varepsilon = 0.1$. Usar o algoritmo de procura de Armijo com backtracking para calcular o η_k, em cada iteração. Considere $\delta = 0.1$.

Um *data set* para análise envolvendo otimização é tipicamente da forma

$$D = \{(x^n, y^n)_{n=1}^N\}$$

- $x^n = (x_1^n, x_2^n, \ldots, x_d^n)$ é o vetor dos atributos (*features*);
- y^n é o rótulo ou etiqueta (*label*), associado ao vetor dos atributos x^n.

A análise consiste em encontrar uma função de previsão $\Phi(w; x)$ que associe features às labels

$$\Phi(w; x^n) \approx y^n, \quad n = 1, \ldots, N$$

em algum sentido ótimo.

1. Ao processo de encontrar $\Phi(w; x)$ é chamado aprendizagem (learning) ou treino (training).
2. Quando y^n é um número real, tem-se um problema de regressão.
3. Quando y^n indica que x^n pertence a uma das M classes ($M \geq 2$), tem-se um problema de classificação.

   $M = 2$, problema de classificação binária

4. As *labels* podem ser nulas, i.é, podem não existirem. Nesse caso, pode-se querer agrupar os atributos x^n em clusters (clusterização) ou identificar um subespaço de menor dimensão onde estão os atributos x^n (redução de dimensionalidade).

Tal análise de dados é muitas vezes referida como aprendizagem automática (*machine learning*). Esta pode ser de dois tipos:

- aprendizagem supervisionada (quando as labels existem)
- aprendizagem não-supervisionada (quando as labels são nulas)

Apresenta-se agora métodos de otimização para aprendizagem automática supervisionada de grande dimensão (*large-scale supervised machine learning*).

Em aprendizagem automática supervisionada, tem-se acesso a um data set de treino $D = \{(x^n, y^n)_{n=1}^N\}$ de features e labels (que pode vir todo de uma só vez ou de forma incremental).

O objetivo é encontrar uma função de previsão Φ. Assume-se que Φ é parametrizada por um vetor real $w \in \mathbb{R}^d$, $\Phi(w; x^n)$, sobre o qual será realizada a otimização.

A previsão incorreta é quantificada por uma **função de perda** ℓ. Isto é, dado um vetor de atributos x^n, o desvio/resíduo entre o valor previsto pela função de previsão $\Phi(w; x^n) = \widetilde{y}^n$ e o valor conhecido y^n, é medido pela função de perda

$$\ell(\Phi(w; x^n), y^n).$$

O objetivo é determinar os valores dos parâmetros $w \in \mathbb{R}^d$ da função de previsão $\Phi(w; x^n)$ que minimizam a função de perda ℓ.

Na prática, o **problema de otimização consiste na minimização do risco empírico** (i.é, a média das perdas do data set de treino D):

$$\underset{w \in \mathbb{R}^d}{\text{minimizar}} \; F(w) = \frac{1}{N} \sum_{n=1}^{N} f^n(w; x^n, y^n) \tag{1}$$

onde

- cada função $f^n(w; x^n, y^n)$ representa $\ell(\Phi(w; x^n), y^n)$

O vetor gradiente da função F é dado por

$$\nabla F(w) = \frac{1}{N} \sum_{n=1}^{N} \nabla f^n(w; x^n, y^n) \tag{2}$$

onde ∇f^n é vetor das primeiras derivadas parciais de f^n

$$\nabla f^n(w) = \begin{pmatrix} \frac{\partial f^n}{\partial w_1} \\ \vdots \\ \frac{\partial f^n}{\partial w_d} \end{pmatrix} \tag{3}$$

- Quando N é elevado, podendo ser da ordem dos milhões ou biliões, o problema (1) é computacionalmente pesado.
- Consequentemente, avaliações de $F(w)$ e $\nabla F(w)$ são de elevado custo computacional.

Método do gradiente estocástico

Existem duas classes principais de algoritmos para machine learning: estocástica e batch. Na primeira classe, um dos métodos mais conhecidos é o método do gradiente estocástico, que no contexto da resolução do problema do (1) é dado por:

Algoritmo 1: Método do Gradiente Estocástico

1. Dar: conjunto de treino $D = \{(x^n, y^n)_{n=1}^N\}$, $w^{(1)} \in \mathbb{R}^d$
2. Fazer $k = 1$
3. **Enquanto** ($w^{(k)}$ não satisfaz o critério de paragem)
4. Gerar aleatoriamente um índice $n_k \in \{1, \ldots, N\}$
5. Calcular o gradiente estocástico $\nabla f^{n_k}(w^{(k)}; x^{n_k}, y^{n_k})$
6. Fazer a direção de procura $s^{(k)} = -\nabla f^{n_k}(w^{(k)}; x^{n_k}, y^{n_k})$
7. Escolher um comprimento do passo $\eta_k > 0$
8. Fazer $w^{(k+1)} = w^{(k)} + \eta_k s^{(k)}$
9. Fazer $k = k + 1$
10. **Fim enquanto**

Observações sobre o método do gradiente estocástico:

- cada iteração é de reduzido custo computacional (calcula apenas um gradiente $\nabla f^{n_k}(w^{(k)}; x^{n_k}, y^{n_k})$ correspondente a um elemento do data set)
- $\{w^{(k)}\}$ é um processo estocástico cujo comportamento é determinado pela sucessão aleatória $\{n_k\}$.
- $-\nabla f^{n_k}(w^{(k)}; x^{n_k}, y^{n_k})$ poderá não ser uma direção de descida para F a partir de $w^{(k)}$ (embora possa ser uma direção de descida *em esperança matemática*);

Método do gradiente batch

Na segunda classe, um dos métodos mais conhecido é o método do gradiente, também designado por gradiente batch ou gradiente completo, que no contexto da resolução do problema do (1) é dado por:

Algoritmo 2: Método do Gradiente batch

1. Dar: conjunto de treino $D = \{(x^n, y^n)_{n=1}^N\}$, $w^{(1)} \in \mathbb{R}^d$
2. Fazer $k = 1$
3. **Enquanto** ($w^{(k)}$ não satisfaz o critério de paragem)
4. Calcular o gradiente $\nabla F(w^{(k)}) = \frac{1}{N} \sum_{n=1}^N \nabla f^n(w^{(k)}; x^n, y^n)$
5. Fazer a direção de procura $s^{(k)} = -\nabla F(w^{(k)})$
6. Escolher um comprimento do passo $\eta^{(k)} > 0$
7. Fazer $w^{(k+1)} = w^{(k)} + \eta_k s^{(k)}$
8. Fazer k=k+1
9. **Fim enquanto**

Observações sobre o método do gradiente batch:

- cada iteração é de maior custo computacional (calcula os N gradientes $\nabla f^n(w^{(k)}; x^n, y^n)$), mas espera-se obter-se melhores passos;
- o método pode beneficiar de paralelização de uma maneira distribuída (devido à estrutura do somatório de F e ∇F);
- o uso do gradiente completo abre as portas para usar todos os métodos de otimização não linear baseados no gradiente.

No entanto, existem razões que levam a que o método do gradiente estocástico seja preferível quando comparado ao método do gradiente batch:

- Muitos conjuntos de treino de grande dimensão envolvem redundância, portanto usar todos os dados em cada iteração de otimização é ineficiente.

  (Imagine se o conjunto de treino D consiste em 10 de cópias de um dado subconjunto D_{sub}. Um minimizante de F usando o conjunto maior D, é claramente dado por um minimizante de F usando o conjunto mais pequeno D_{sub}. Se fosse aplicado um método batch para minimizar F usando D, então cada iteração seria 10 vezes mais dispendiosa do que se tivesse usado apenas uma cópia D_{sub}. Por outro lado, o método do gradiente estocástico realiza os mesmos cálculos em ambos os cenários.)

- O método do gradiente estocástico normalmente faz uma melhoria inicial bastante rápida, seguida de uma desaceleração após 1 ou 2 épocas (1 época = N avaliações consecutivas do gradiente estocástico).

  (Para se ter tal comportamento eficiente do método é necessário usar um bom comprimento do passo. Em geral, para resolver o problema em mãos, é necessário executar o método usando diferentes valores de η e identificar o melhor η.)

Método do gradiente estocástico mini-batch

O método do gradiente estocástico mini-batch, é uma variante do método do gradiente, o qual usa um subconjunto de gradientes estocásticos, por iteração.

Algoritmo 3: Método do gradiente estocástico mini-batch

1. Dar: conjunto de treino $D = \{(x^n, y^n)_{n=1}^N\}$, $w^{(1)} \in \mathbb{R}^d$
2. Fazer $k = 1$
3. **Enquanto** ($w^{(k)}$ não satisfaz o critério de paragem)
4. Gerar aleatoriamente um subconjunto de índices $D_k \subseteq \{1, \ldots, N\}$
5. Calcular o gradiente $\nabla F_{D_k}(w^{(k)}) = \frac{1}{|D_k|} \sum_{n \in D_k} \nabla f^n(w^{(k)}; x^n, y^n)$
6. Fazer a direção de procura $s^{(k)} = -\nabla F_{D_k}(w^{(k)})$
7. Escolher um comprimento do passo $\eta_k > 0$
8. Fazer $w^{(k+1)} = w^{(k)} + \eta_k s^{(k)}$
9. **fim enquanto**

Observações sobre o método do gradiente estocástico mini-batch:

- o método pode beneficiar de paralelização de uma maneira distribuída (devido à estrutura do somatório de ∇F_{D_k});
- o uso de mais gradientes estocásticos por iteração, faz com que o método seja mais fácil de afinar em termos de η.

Problema: Considere-se seguinte problema de *machine learning*. Dado o data set $D = (x^n, y^n)_{n=1}^N$ pretende-se determinar os coeficientes de um polinómio de grau I

$$\begin{aligned} \phi(w; x) = & \quad w_0 + w_1 x + w_2 x^2 + \cdots + w_{\mathrm{I}} x^{\mathrm{I}} \\ = & \quad w^T p(x) \end{aligned}$$

onde $p(x) = (1, x, x^2, \ldots, x^{\mathrm{I}})^T$ que melhor ajustam o polinómio aos dados D no sentido da minimização da função custo MSE (*Mean Squared error*):

$$MSE(w; D) = \frac{1}{N} \sum_{n=1}^{N} (\phi(w; x^n) - y^n)^2.$$

Nota: O gradiente da função $MSE(w; D)$ é dado por

$$\nabla MSE(w; D) = \frac{2}{N} \sum_{n=1}^{N} (\phi(w; x^n) - y^n) p(x^n).$$

Exercício 3:

Resolver o Problema apresentado com o data set data1.csv (N=100).

- Dividir o data set em duas partes: 80% para treino D_t e 20% para validação D_v. Esta selecção deverá ser aleatória.
- Implementar o método do gradiente batch (**Algoritmo 2**) e como aproximações iniciais aos parâmetros considere $w^{(1)} = (0, 0, \ldots, 0)$.
- Use como critério de paragem

$$\|\nabla MSE(w)\| \leq 10^{-4} \text{ e } k \leq 10N_t.$$

- Calcular o comprimento do passo η pelo algoritmo de procura de Armijo com backtracking e faça $\delta = 0.1$.
- Resolva o Problema considerando polinómios de grau $\mathrm{I} = 2, \ldots, 7$.
- Para cada um dos polinómios calcule: w^*, o erro de treino (*in-sample error*) $MSE(w^*; D_t)$, e o erro de validação (*out-sample error*) $MSE(w^*; D_v)$. Fazer o gráfico dos erros e indicar qual o grau I que fornece a melhor aproximação.

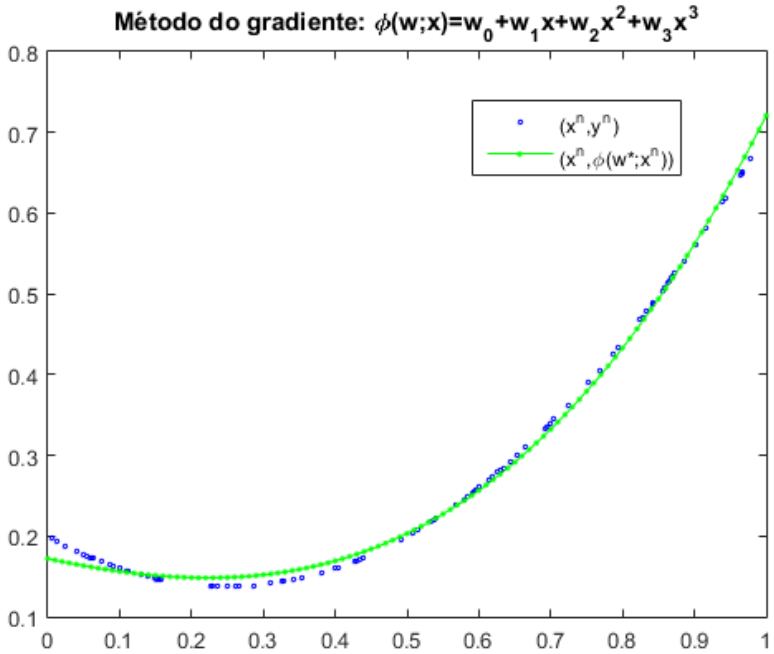
Exercício 4: Faça o exercício 3 mas considere a implementação do método do gradiente estocástico (**Algoritmo 1**). Notar que terá que adaptar a condição de Armijo a este caso.

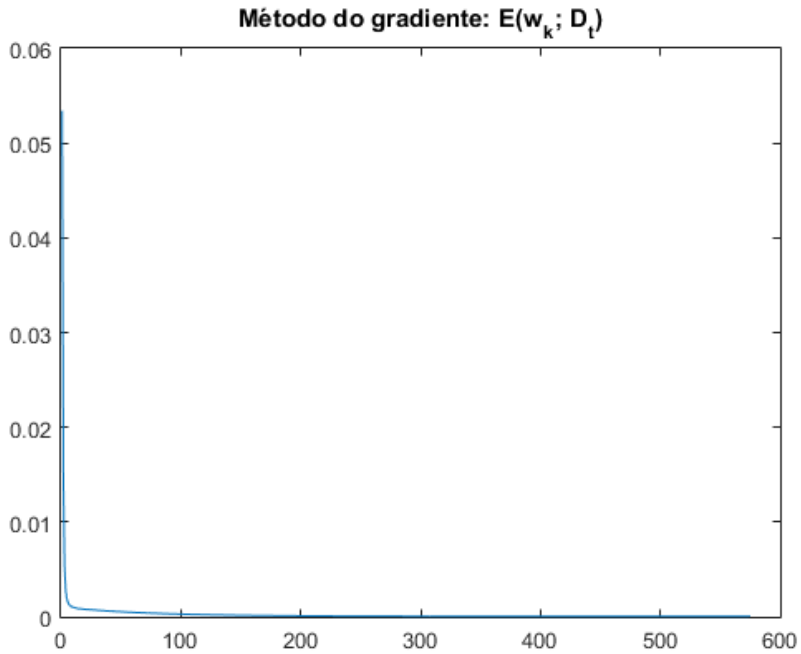
Exercício 5: Faça o exercício 3 mas considere a implementação do método do gradiente estocástico mini-batch (**Algoritmo 3**). Considere para mini-batch 5% dos dados do D_t. Esta selecção deverá ser aleatória em cada iteração. Notar que terá que adaptar a condição de Armijo a este caso.

Exercício 6:

1. Avaliar o desempenho dos algoritmos desenvolvidos com: η constante ao longo processo iterativo (exemplo: $\eta = 1$, $\eta = 0.1$, ...).
2. Avaliar o desempenho dos algoritmos 1 e 3 quando: $\eta = 0.1$ e reduz para 0.01, 0.001, ... de $\times$ em $\times$ épocas.

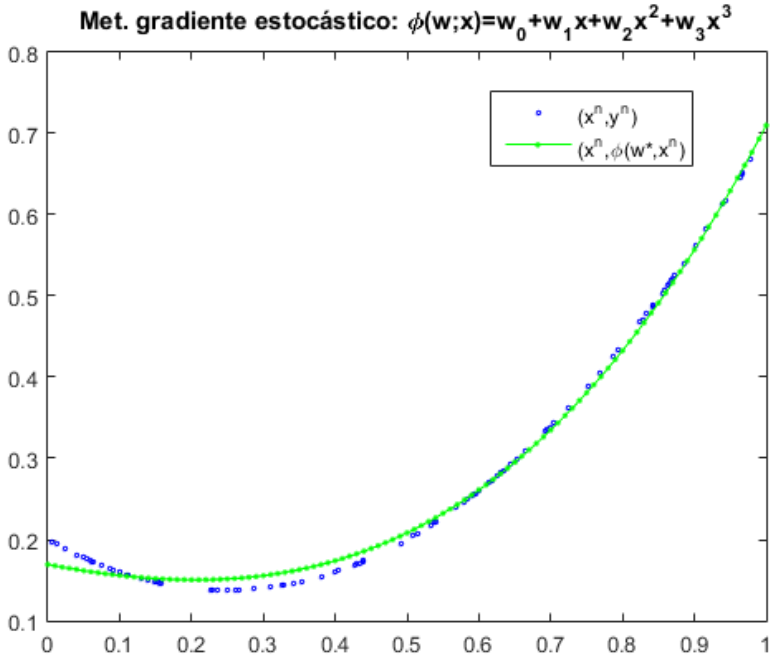
Exercício 3: Solução com I = 3

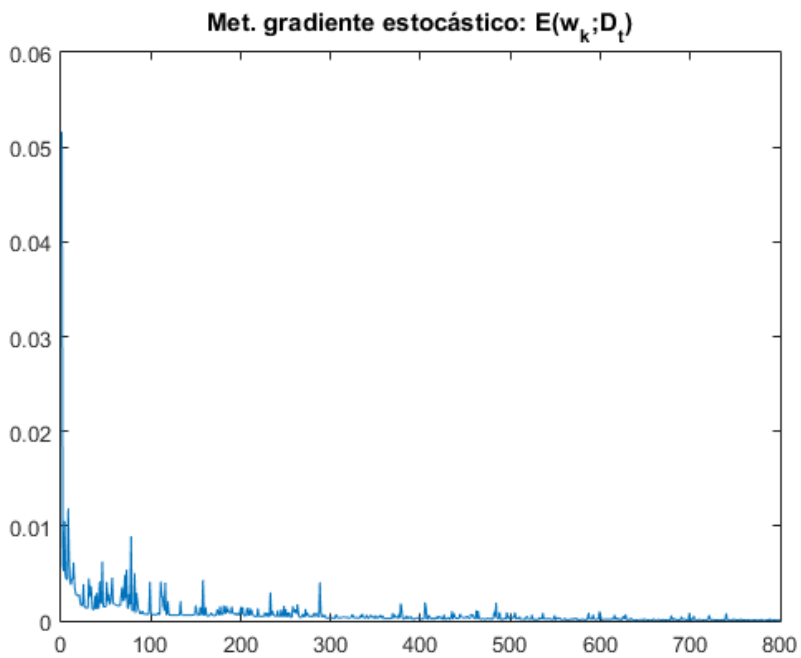




- $w^* = (0.1763, -0.2063, 0.2817, 0.4703)$
- $MSE(w^*, D_t) = 3.5315 \times 10^{-5}$
- $MSE(w^*, D_v) = 3.5042 \times 10^{-5}$

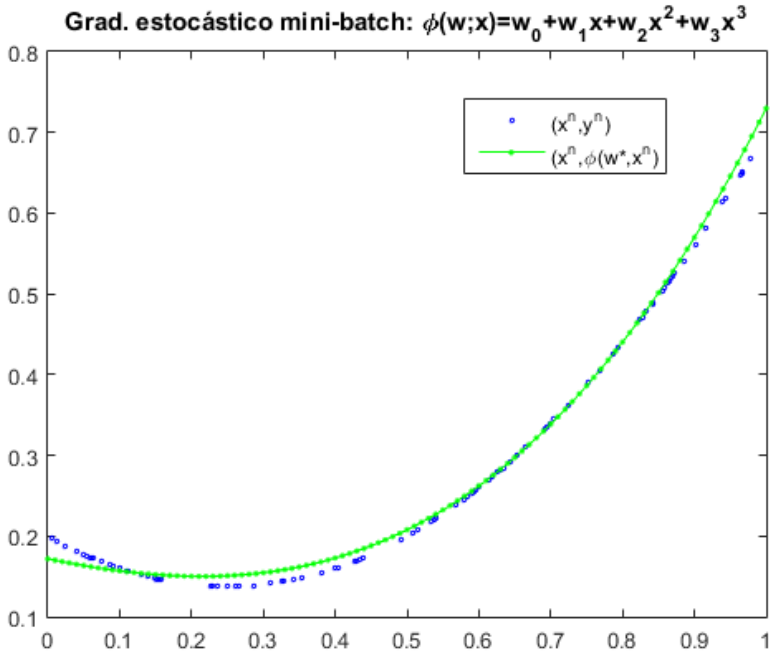
Exercício 4: Solução com I = 3

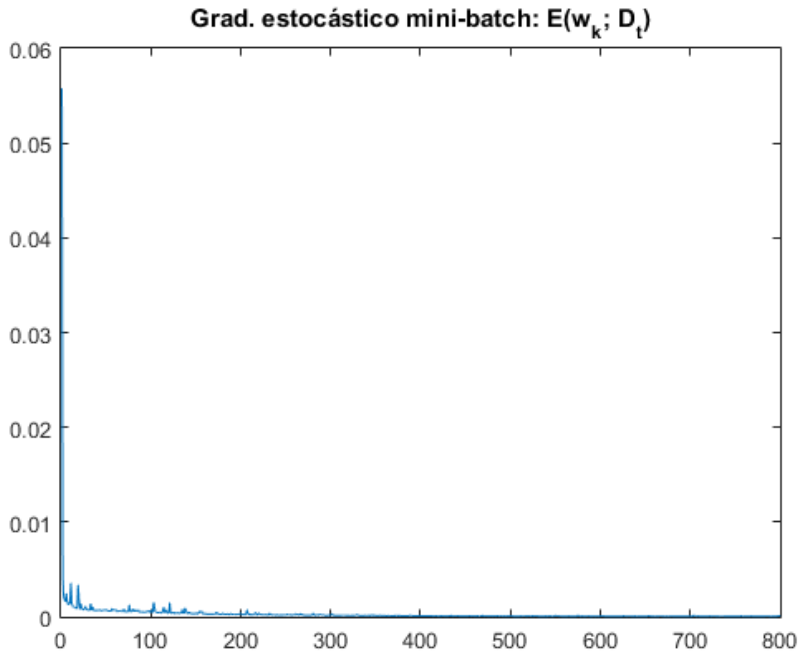




- $w^* = (0.1693, -0.1657, 0.2681, 0.4384)$
- $MSE(w^*, D_t) = 5.1689 \times 10^{-5}$
- $MSE(w^*, D_v) = 3.2782 \times 10^{-5}$

Exercício 5: Solução com I = 3





- $w^* = (0.1718, -0.1818, 0.2747, 0.4652)$
- $MSE(w^*, D_t) = 5.3218 \times 10^{-5}$
- $MSE(w^*, D_v) = 4.9128 \times 10^{-5}$

Bibliografia

- Jorge Nocedal and Stephen Wright. **Numerical Optimization**, Second Edition, Springer 2006
- Léon Bottou, Frank E. Curtis and Jorge Nocedal. **Optimization Methods for Large-Scale Machine Learning**, Northwestern University, 2016.