

T319 - Introdução ao Aprendizado de Máquina: *Regressão Linear (Parte I)*



Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Motivação

- **Exemplo 1:** Estimar o preço de casas.
- **Exemplo 2:** Estimar as vendas de sorvete.

500 m²



R\$ 1.000.000,00

70 m²



R\$ 200.000,00

200 m²



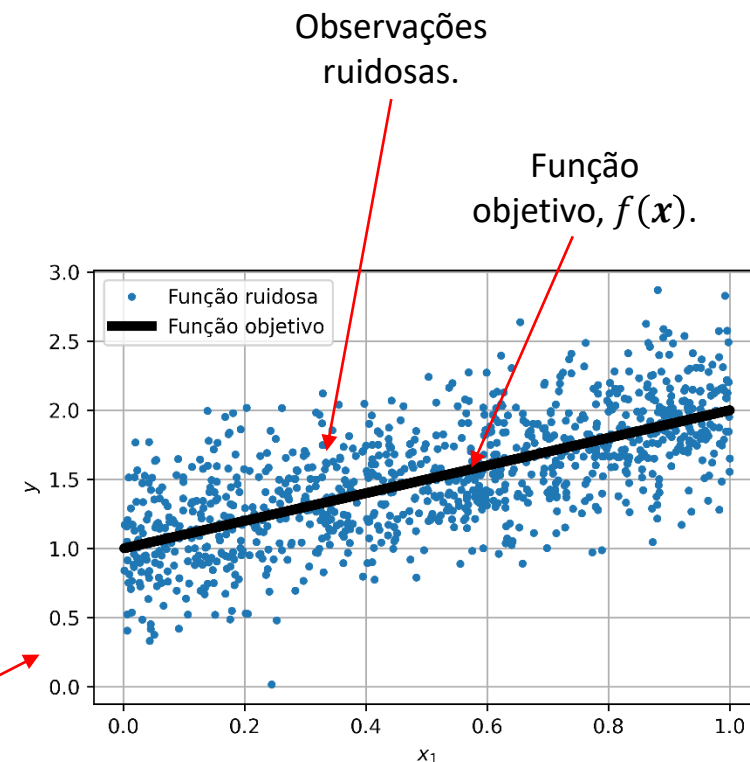
???



- Podemos encontrar uma relação matemática, $f(x)$, entre a área, localização, n° de quartos de uma casa e seu valor?
- Ou entre a temperatura e a quantidade de sorvetes vendidos?

Regressão Linear

- É um dos mais antigos e conhecidos algoritmos de aprendizado de máquina.
- Vai nos dar várias **intuições** importantes para o **entendimento** de outros **algoritmos mais complexos**, como, por exemplo, **classificadores** e **redes neurais**.
- **Objetivo**: encontrar uma função, $h(x)$, que **mapeie**, de **forma ótima**, os **atributos**, x , em uma variável de saída $\hat{y} = h(x)$, de tal forma que $h(x)$ seja uma boa aproximação da **função verdadeira** ou **objetivo**, $f(x)$.
- $f(x)$ é muitas vezes **desconhecida** e temos acesso apenas a **observações ruidosas**.
- Regressão também é conhecida como **aproximação de funções** ou **ajuste de curvas**.
- Como encontramos uma função, $h(x)$, que **aproxime** $f(x)$ de forma ótima a partir de dados ruidosos?

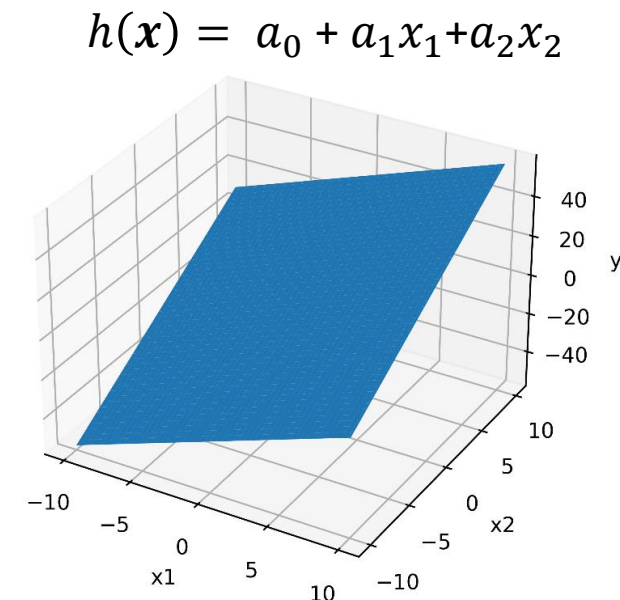
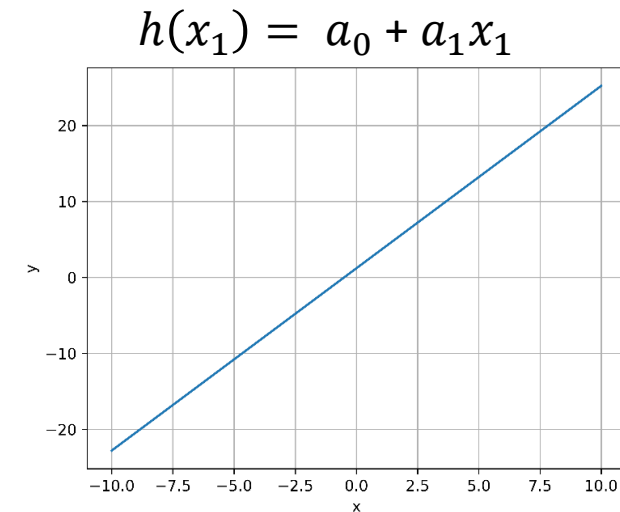


Temos x (atributos) e y (rótulos) e queremos encontrar $\hat{y} = h(x)$.

Que tipo de aprendizado?

Regressão Linear

- **Qual forma deve ter a função $h(x)$?** Os modelos mais simples são:
 - Com apenas um atributo, x_1 , $h(x)$ é uma reta, $h(x) = a_0 + a_1x_1$.
 - Com dois atributos, x_1 e x_2 , $h(x)$ é uma superfície 2D (ou seja, um plano), $h(x) = a_0 + a_1x_1 + a_2x_2$.
 - E assim por diante.
- **Modelo geral:** equação de um **hiperplano**
$$h(x) = a_0 + a_1x_1 + \dots + a_Kx_K = a_0 + \sum_{i=1}^K a_i x_i.$$
- Existem outros modelos, os quais veremos mais adiante.
- Na literatura, a função $h(x)$, é chamada de **função hipótese**, pois é uma das **possíveis soluções** encontradas no **espaço de hipóteses**, H , para aproximar $f(x)$.
- **Espaço de hipóteses:** é conjunto de todas as possíveis **funções hipótese**.
 - Superfície formada por todos os possíveis valores dos parâmetros, a_k , $\forall k$ quando substituídos em $h(x)$.

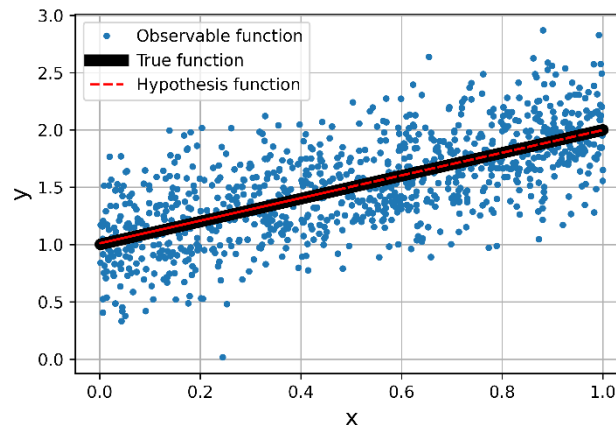


Regressão Linear

- Agora que temos uma forma para $h(\mathbf{x})$, podemos refinar o objetivo da regressão um pouco mais.
- **Objetivo: Encontrar os parâmetros**, também chamados de **pesos**, a_0, a_1, \dots, a_K de tal forma que $h(\mathbf{x})$ seja uma **aproximação ótima** de $f(\mathbf{x})$.
 - **Ótima** no sentido que **minimiza** uma métrica de erro que iremos definir a seguir.
- **Aprendizado supervisionado**: atributos, \mathbf{x} , mais rótulos/objetivos, y .
- A **regressão** é chamada de **linear** porque a variável de saída, y , é modelada como sendo uma **combinação linear** dos **atributos**, \mathbf{x} .
 - **OBS.: Linear**, nesse contexto, significa “*linear com relação aos pesos*” e não com relação aos **atributos**, i.e., \mathbf{x} . Desta forma, os seguintes **modelos** também são lineares com relação aos pesos:
 - ✓ $h(\mathbf{x}) = a_0 + a_1 \log x_1 + a_2 \cos x_2$
 - ✓ $h(\mathbf{x}) = a_0 + a_1 e^{x_1}$
 - ✓ $h(\mathbf{x}) = a_0 + a_1 x_1^2$
- Exemplo de um modelo não-linear: $h(\mathbf{x}) = \frac{a_0 x_1}{a_1 + x_1}$.

Não é possível expressar a equação como uma combinação linear dos atributos.

Definição formal do problema



O problema da **regressão linear** pode ser definido da seguinte forma:

- **Dados disponíveis:**

- Conjunto de N observações (conjunto de pares de treinamento) : $\{x(i), y(i)\}, i = 0, \dots, N - 1$, onde
 - $x(i) \in \mathbb{R}^{K+1 \times 1}$: i -ésimo vetor de atributos de entrada com dimensão $K + 1 \times 1$, ou sejam $K + 1$ atributos.
 - $y(i) \in \mathbb{R}$: i -ésimo valor esperado de saída referente ao vetor de entrada $x(i)$.

Atributo de bias,
sempre igual a 1.

- **Modelo:**

$$\hat{y}(i) = h(x(i)) = a_0 + a_1 x_1(i) + \dots + a_K x_K(i) = \mathbf{a}^T \mathbf{x}(i),$$

onde $\mathbf{a} = [a_0, \dots, a_K]^T$ e $\mathbf{x}(i) = [1, x_1(i), \dots, x_K(i)]^T$.

- \mathbf{a} é um vetor **coluna** com dimensão $(K + 1 \times 1)$ contendo os **pesos** da **função hipótese** e $x(i)$ é um vetor coluna com dimensão $(K + 1 \times 1)$ contendo os i -ésimos valores dos **atributos**.
- a_0 é o **coeficiente linear**, ou seja, é o valor de $h(x)$ que intercepta o eixo das ordenadas, y , a_0 é conhecido também como **intercept** ou **bias**.
- Como a_0 não tem um **atributo** relacionado a ele, para facilitar o modelamento matemático, criamos um atributo falso, x_0 , com valor constante sempre igual a 1, i.e., $x_0 = 1$.
- **Objetivo:** encontrar um vetor de pesos \mathbf{a} que minimize o **erro**, dado por uma **função de erro**, $J_e(\mathbf{a})$, entre a aproximação $\hat{y}(i)$ e o valor esperado $y(i)$ para **todos os exemplos do conjunto**.
- Ou seja, o treinamento do modelo envolve a minimização de uma **função de erro**.

$$\min_{\mathbf{a}} J_e(\mathbf{a})$$

- Portanto, precisamos definir uma **função de erro**.

Função de Erro

- **Função de erro:** existem várias possibilidades para se definir a **função de erro** a ser minimizada, porém, geralmente, utiliza-se o **erro quadrático médio**

$$J_e(\mathbf{a}) = \frac{1}{N} \sum_{i=0}^{N-1} (y(i) - \hat{y}(i))^2 = \frac{1}{N} \sum_{i=0}^{N-1} (y(i) - h(\mathbf{x}(i), \mathbf{a}))^2,$$

Erro entre a saída da função hipótese e a saída esperada.

que nada mais é do que a **média aritmética do quadrado dos erros**.

- Nós veremos mais adiante a razão pela qual o **erro quadrático médio** é utilizado.
- A **função de erro** pode ser reescrita em **forma matricial** como

$$J_e(\mathbf{a}) = \frac{1}{N} \|\mathbf{y} - \Phi \mathbf{a}\|^2,$$

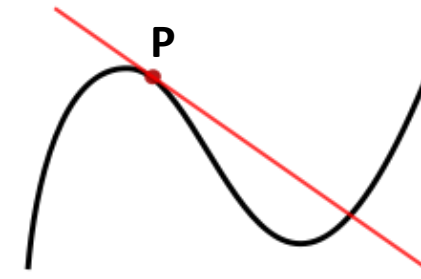
onde $\mathbf{y} = [y(0), \dots, y(N-1)]^T$ é um vetor $(N \times 1)$, $\Phi = [\mathbf{x}(0), \dots, \mathbf{x}(N-1)]^T$ é uma matriz $(N \times K+1)$ e N é o número de exemplos ou observações.

- Então, para encontrarmos o **vetor de pesos**, \mathbf{a} , devemos minimizar a função de erro:

$$\min_{\mathbf{a} \in \mathbb{R}^{K+1 \times 1}} \|\mathbf{y} - \Phi \mathbf{a}\|^2.$$

OBS.: Por ser constante, $1/N$ não influencia na minimização e, portanto, pode ser omitido.

Minimizando a Função de Erro



A reta vermelha é **tangente** à curva no ponto P (vermelho). Sua **inclinação** é a **derivada** da função naquele ponto.

Como encontramos o mínimo da função de erro em relação aos pesos?

- Da disciplina de cálculo, sabemos que derivando a **função de erro**, $\|y - \Phi a\|^2$, com relação ao vetor a e igualando a 0, nós encontramos o **ponto** onde a **inclinação** de uma **reta tangente** à **função de erro** é nula:

$$\frac{\partial \|y - \Phi a\|^2}{\partial a} = 0 \quad \therefore 2a^T \Phi^T \Phi - 2y^T \Phi = 0,$$

porém, esse **ponto** pode ser tanto um **mínimo** quanto um **máximo** da **função de erro**, pois em ambos os pontos a **inclinação** da reta tangente é **nula**.

Então, como sabemos se o ponto encontrado é um mínimo ou um máximo?

- Se a inclinação da tangente é nula e a **derivada de segunda ordem** da função for **positiva**, então o ponto nos dá o mínimo da função,

$$\frac{\partial^2 \|y - \Phi a\|^2}{\partial^2 a} = 2\Phi^T \Phi.$$

- Se a matriz Φ tiver **posto** igual a $K + 1$, então a matriz $\Phi^T \Phi$ é **positiva semi-definida** e, portanto, o ponto encontrado acima é realmente o ponto de mínimo da **função de erro**.
 - **Posto de uma matriz**: é o número de linhas ou colunas linearmente independentes da matriz.
 - Uma matriz quadrada $\Phi^T \Phi$ é **positiva semi-definida** se $x^T \Phi^T \Phi x = \|\Phi x\|^2 \geq 0, \forall x \neq 0$.

Minimizando a Função de Erro

- Portanto, voltando à equação da derivada parcial de primeira ordem igual a 0, temos

$$\mathbf{a}^T \Phi^T \Phi = \mathbf{y}^T \Phi.$$

- Após aplicarmos o transposto a ambos os lados e isolar \mathbf{a} temos

$$\mathbf{a} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}.$$

- Essa equação é conhecida como **equação normal** e nos dá a **solução ótima** em relação a minimização do **erro quadrático médio** para esse **sistema de equações lineares**.

Observações:

1. O método encontra uma **solução única** se e somente se a matriz, $\Phi^T \Phi$, for **invertível** (i.e., se ela for **não-singular**), ou seja, com **posto** igual a $K + 1$.
2. O método só funciona para sistemas **determinados** ou **sobredeterminados**, ou seja, quando o número de equações (i.e., pares x e y) é **igual ou maior** do que o número de incógnitas (i.e., pesos), ou seja, $N \geq K + 1$.
3. Para sistemas **subdeterminados**, ou seja, que têm menos equações do que incógnitas, a matriz $\Phi^T \Phi$ tem **posto** menor do que $K + 1$ e, portanto, é **singular** (ou seja, a matriz $\Phi^T \Phi$ não tem uma inversa). Neste caso, não existe solução ou ela não é única.

[Exemplo: normal equation example1.ipynb](#)

Superfície de Erro

- E se plotarmos a função de erro, $J_e(\mathbf{a})$, em função dos pesos, \mathbf{a} ?

$$J_e(\mathbf{a}) = \frac{1}{N} \sum_{i=0}^{N-1} (y(i) - \hat{y}(i))^2 = \frac{1}{N} \sum_{i=0}^{N-1} (y(i) - h(\mathbf{x}(i), \mathbf{a}))^2.$$

- Que forma vocês acham que ela terá?
- $J_e(\mathbf{a})$ faz o **mapeamento** entre o **vetor de pesos** e o erro correspondente:
 - $J_e(\mathbf{a}): \mathbb{R}^{K+1 \times 1} \rightarrow \mathbb{R}$. Esse mapeamento define o que conhecemos como **superfície de erro**.
- Se expandirmos $J_e(\mathbf{a})$ notamos que ela possui forma **quadrática** com respeito ao **vetor de pesos**, \mathbf{a} .

$$J_e(\mathbf{a}) = \|\mathbf{y} - \Phi \mathbf{a}\|^2 = \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \Phi \mathbf{a}^T - \mathbf{a}^T \Phi^T \mathbf{y} + \boxed{\mathbf{a}^T \Phi^T \Phi \mathbf{a}}.$$

Termo quadrático, $\|\Phi \mathbf{a}\|^2$.

- Consequentemente, a superfície é **convexa** (i.e., tem forma de **tigela**) e, portanto, possui um único **mínimo (global)**, que é encontrado, por exemplo, pela **equação normal**.
- Este é o motivo de usarmos o **erro quadrático médio** como **função de erro**.

Superfície de Erro: Exemplo

- Vamos supor a seguinte **função observável**

$$y_{\text{noisy}}(n) = y(n) + w(n),$$

onde $w(n) \sim N(0, 1)$ e $y(n)$ é a **função objetivo** (ou **modelo gerador**) dada por

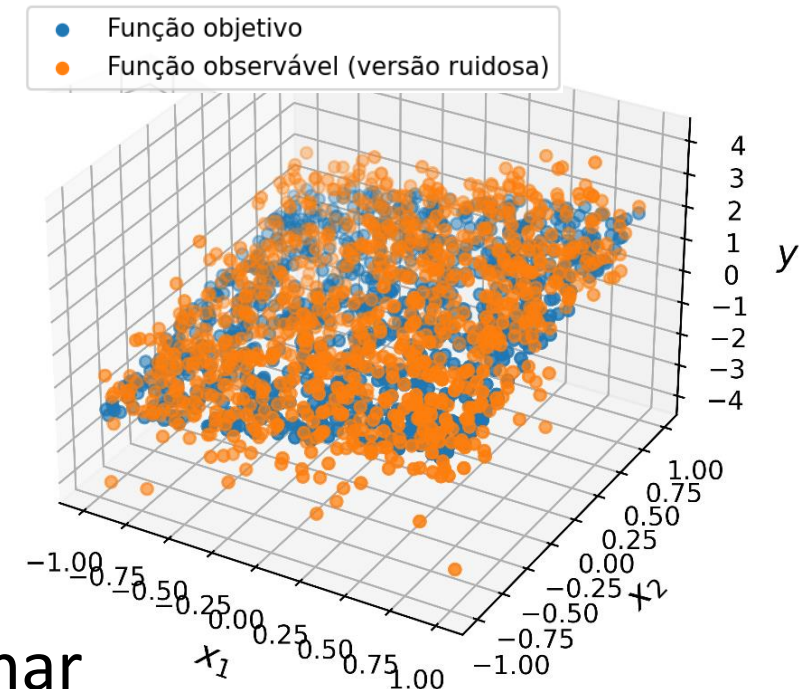
$$y(n) = a_1 x_1(n) + a_2 x_2(n),$$

onde $x_1(n)$ e $x_2(n) \sim U(-1, 1)$ e $a_1 = a_2 = 1$.

- Agora, suponhamos que nós quiséssemos aproximar a **função objetivo** a partir, apenas, de suas **amostras ruidosas** com a seguinte **função hipótese**

$$h(\mathbf{x}, n) = \hat{y}(n) = \hat{a}_1 x_1(n) + \hat{a}_2 x_2(n).$$

- Como encontraríamos os valores de \hat{a}_1 e \hat{a}_2 ?



Comparação da **função objetivo** com sua versão ruidosa.

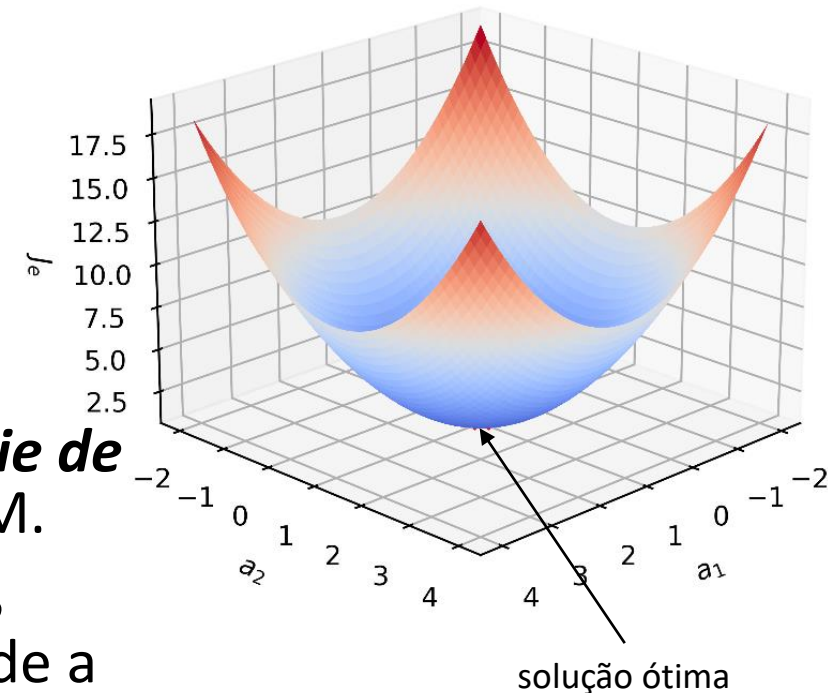
OBS.: Se tivéssemos mais de dois atributos, já não seria possível plotar uma figura.

Superfície de Erro: Exemplo

- Até o momento, conseguiríamos encontrar com a **equação normal** ou **visualmente**, plotando a **superfície de erro** a partir da função do **erro quadrático médio (EQM)**:

$$J_e(a_1, a_2) = \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - \hat{y}(n) \right)^2$$
$$= \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - (\hat{a}_1 x_1(n) + \hat{a}_2 x_2(n)) \right)^2.$$

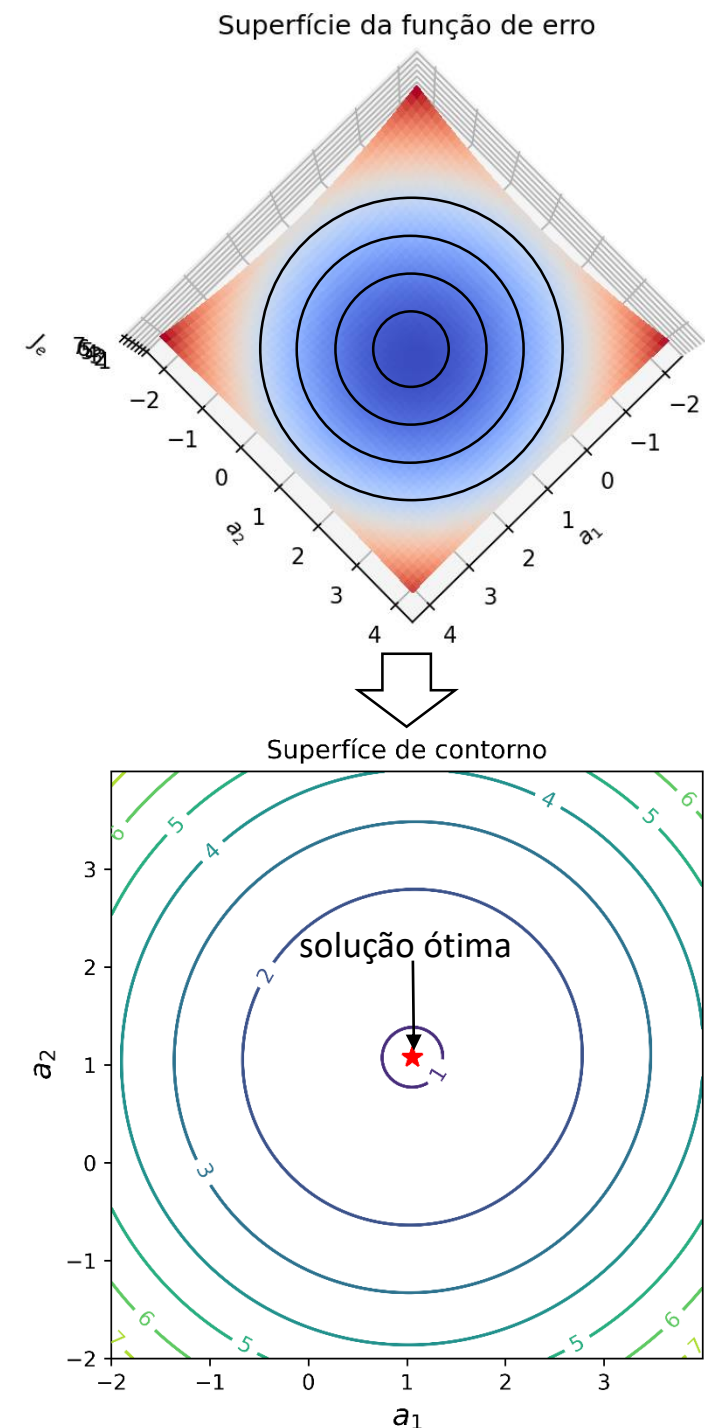
- Os valores de erro, $J_e(a_1, a_2)$, para plotarmos a **superfície de erro** são obtidos variando-se \hat{a}_1 e \hat{a}_2 na equação do EQM.
- A **superfície de erro** é representada por uma figura em 3 dimensões, onde cada par de valores \hat{a}_1 e \hat{a}_2 corresponde a um erro, $J_e(a_1, a_2)$.
- Percebamos que devido a superfície ser **convexa**, temos apenas **um ponto de mínimo**, o **mínimo global**.



Superfície de Erro: Exemplo

- Outra figura importante que podemos plotar a partir dos resultados obtidos para plotarmos a superfície de erro é chamada de **superfície de contorno**.
- Uma **linha de contorno** é uma **curva ao longo da qual a função tem um valor constante**.
- No da superfície de erro, cada uma das linhas indica uma curva ao longo da qual o **erro** é **constante**.
- Ou seja, qualquer par de valores \hat{a}_1 e \hat{a}_2 ao longo de uma curva terá o mesmo valor de erro.

[Exemplo: error_surface_example2.ipynb](#)



Formatos diferentes para a superfície de erro

Seja a seguinte **função objetivo**:

$$y(n) = a_1 x_1(n) + a_2 x_2(n),$$

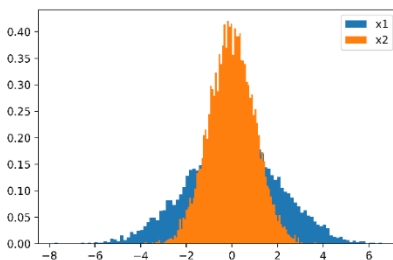
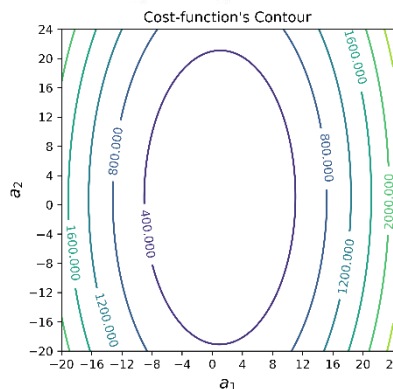
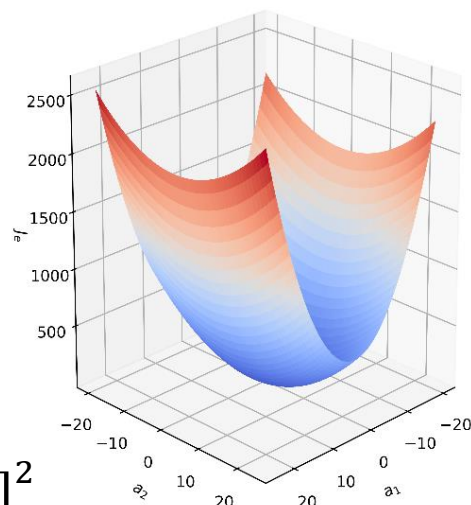
onde $a_1 = 1$, $a_2 = 1$ e **função observável**

$$y_{\text{noisy}}(n) = y(n) + w(n).$$

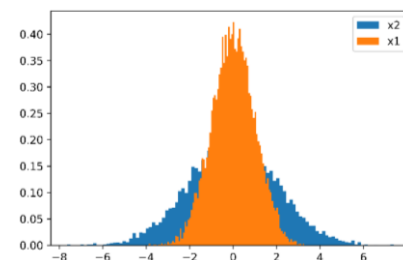
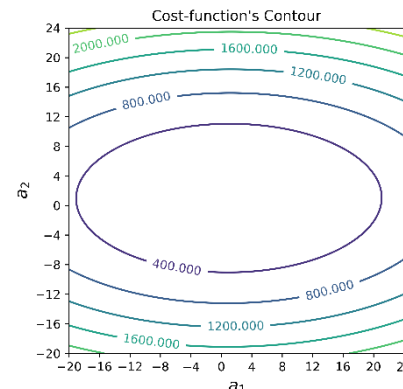
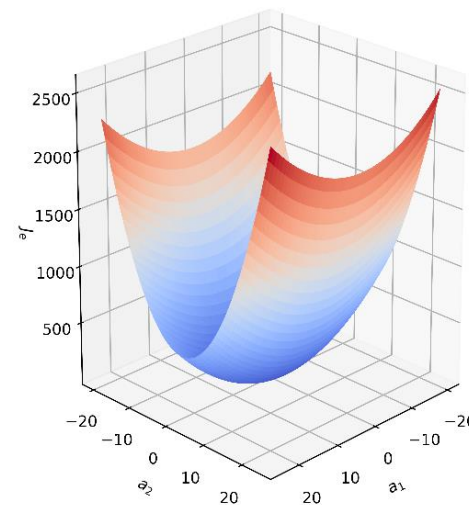
Para plotar a superfície de erro usamos:

$$J_e(\mathbf{a}) = \frac{1}{N} \sum_{n=0}^{N-1} [y_{\text{noisy}}(n) - (\hat{a}_1 x_1(n) + \hat{a}_2 x_2(n))]^2$$

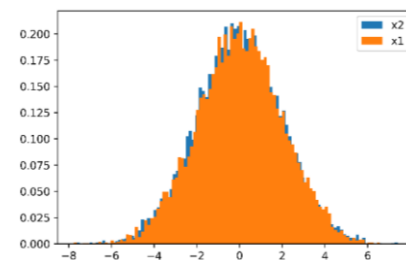
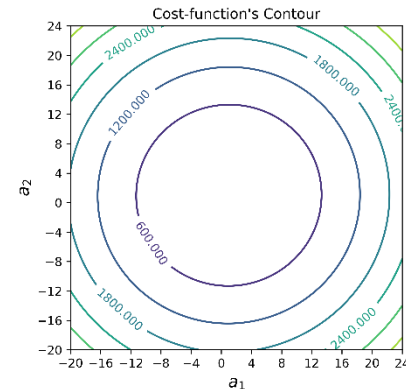
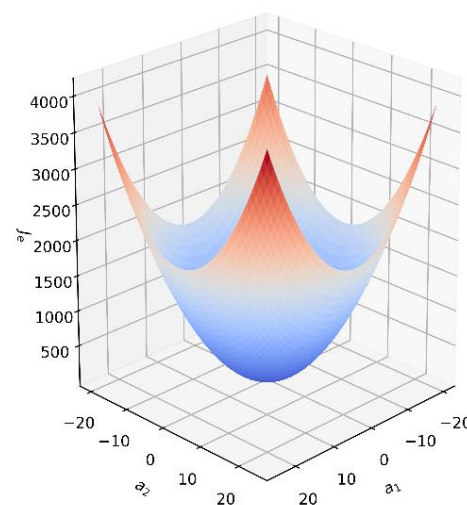
- Se x_1 varia entre um intervalo muito grande de valores, então o **peso** da variação de \hat{a}_1 no erro é maior, ou seja, o erro varia mais rapidamente com variações de \hat{a}_1 , resultando num vale.
- A mesma coisa pode ser dita para x_2 e \hat{a}_2 (vale).
- Quando x_1 e x_2 têm intervalos semelhantes, então, a variação tanto de \hat{a}_1 quanto de \hat{a}_2 tem **pesos** semelhante na variação do erro (tigela).



$x_1 = 2 * \text{randn}(N, 1)$
 $x_2 = \text{randn}(N, 1)$



$x_1 = \text{randn}(N, 1)$
 $x_2 = 2 * \text{randn}(N, 1)$



$x_1 = 2 * \text{randn}(N, 1)$
 $x_2 = 2 * \text{randn}(N, 1)$

[Exemplo: formatos diferentes da superfície de erro.ipynb](#)

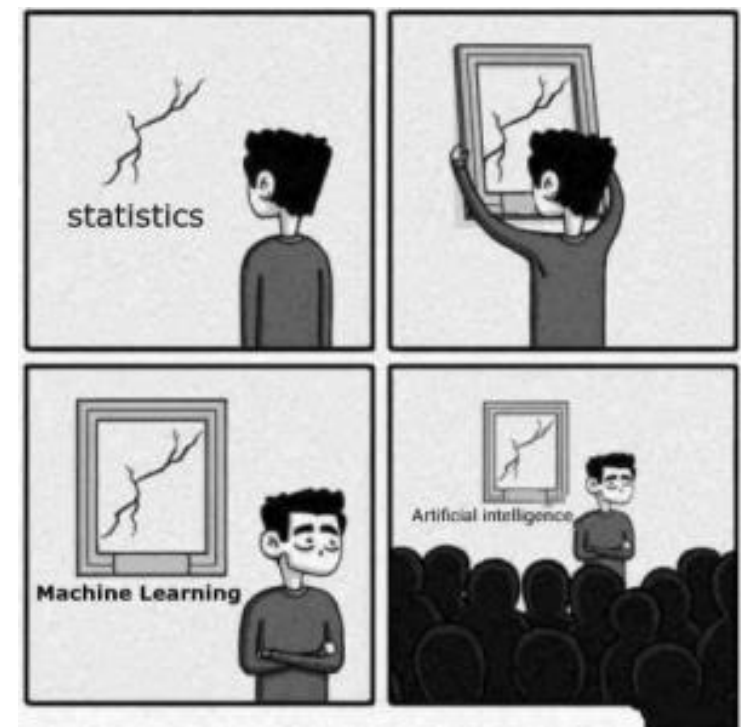
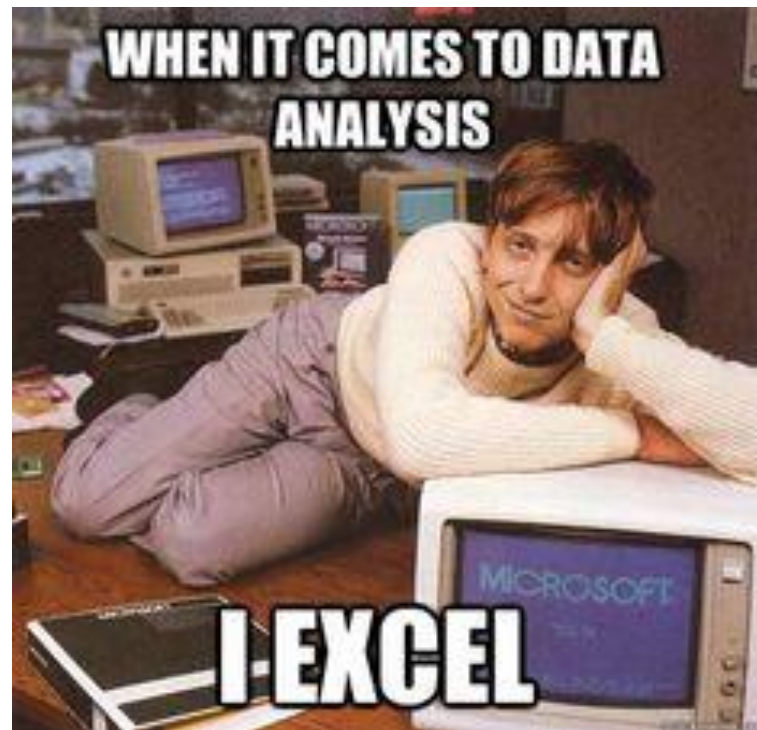
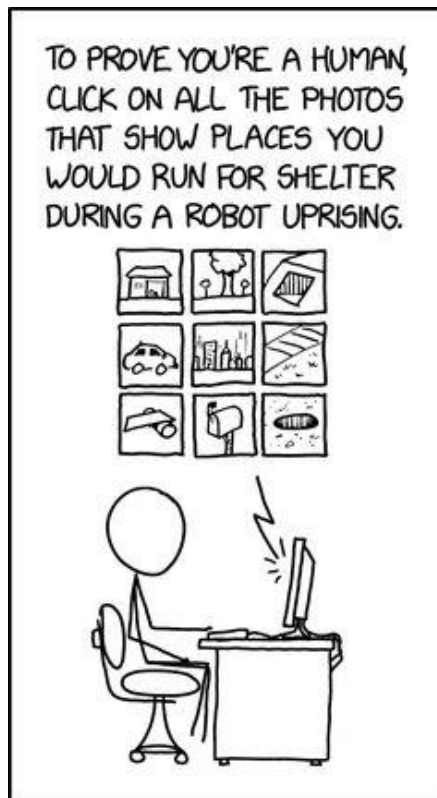
Desvantagens da forma fechada (Eq. Normal)

- **Alta complexidade computacional:** a solução da **equação normal** envolve o cálculo da inversa de $\Phi^T \Phi$, o qual tem complexidade computacional que varia de $O(K^{2.4})$ a $O(K^3)$.
 - **Exemplo:** Se o número de **atributos**, K , dobrar, o tempo para cálculo aumenta de $2^{2.4} = 5.3$ a $2^3 = 8$ vezes.
- Dependendo do número de **exemplos**, N , e de **atributos**, x , a matriz Φ pode consumir muita memória.
- **Portanto, essa abordagem não é escalonável!**
- Adicionalmente, para irmos além dos modelos lineares (i.e., modelos não-lineares como classificadores e redes neurais) precisamos lidar com o fato de que não existem formas fechadas como a **equação normal**.
- **Solução:** abordagens iterativas.
 - São métodos que “**procuram**” de forma iterativa os pesos ótimos no espaço de soluções.
 - Por exemplo, o algoritmo do **gradiente descendente**, o qual veremos a seguir.

Tarefas

- **Quiz:** “*T319 - Quiz - Regressão: Parte I*” que se encontra no MS Teams.
- **Exercício Prático:** [Laboratório #2](#).
 - Pode ser baixado do MS Teams ou do GitHub.
 - Pode ser respondido através do link acima (na nuvem) ou localmente.
 - [Instruções para resolução e entrega dos laboratórios](#).
 - **Laboratórios podem ser feitos em grupo, mas as entregas devem ser individuais.**

Obrigado!



Albert Einstein: Insanity Is Doing
the Same Thing Over and Over Again
and Expecting Different Results

Machine learning:

