Inteligência Artificial
Aula 10 - vídeo 2 - Regressão Linear e Logística

João C. P. da Silva

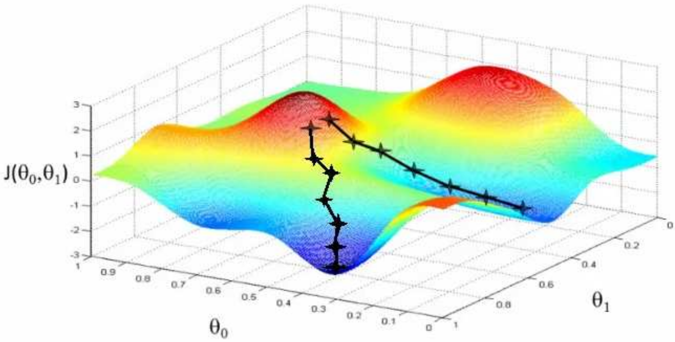
Dept. Ciência da Computação - UFRJ

28 de outubro de 2020

Gradiente Descendente

- Algorítmo mais geral, usado em outros algorítmos de aprendizado de máquina.
- Usado para minimizar vários tipos de função.
- Considere uma função (qualquer) $J(\theta_0, \theta_1)$,onde estamos interessados em $min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$.
    - Inicie com algum θ_0, θ_1 (por exemplo, $\theta_0 = \theta_1 = 0$)
    - Altere os valores de θ_0, θ_1 para reduzir $J(\theta_0, \theta_1)$ até atingir um mínimo.

Gradiente Descendente

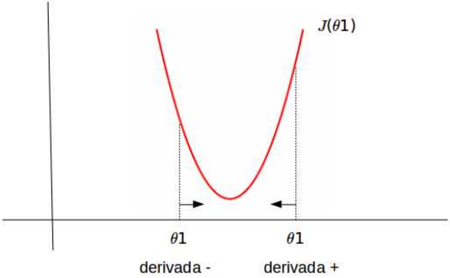


Gradiente Descendente

- Repetir até convergir:
  - $temp_0 := \theta_0 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$
  - $temp_1 := \theta_1 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$
  - $\theta_0 := temp_0$
  - $\theta_1 := temp_1$
- α: taxa de aprendizado (learning rate). Determina o *tamanho do passo* que estamos dando.
- Note que θ_0 e θ_1 são atualizados simultaneamente.

Gradiente Descendente

Considere que queremos $min_{\theta_1} J(\theta_1)$.



$\theta_1 := \theta_1 - \alpha \frac{dJ(\theta_1)}{d\theta_1}$

- $\frac{dJ(\theta_1)}{d\theta_1} > 0 : \theta_1 := \theta_1 - \alpha(+) = \theta_1 - \alpha(\cdots)$
- $\frac{dJ(\theta_1)}{d\theta_1} < 0 : \theta_1 := \theta_1 - \alpha(-) = \theta_1 + \alpha(\cdots)$

Gradiente Descendente para Regressão Linear

- **Regressão Linear**
  - $Y = \theta_0 + \theta_1 * X$
  - $J(\theta_0, \theta_1) = \sum_{i=1}^{n} (y_{i_{real}} - y_{i_{previsto}})^2 = \sum_{i=1}^{n} (y_i - (\theta_1 * x_i + \theta_0))^2$

Gradiente Descendente para Regressão Linear

- **Regressão Linear**
  - $Y = \theta_0 + \theta_1 * X$
  - $J(\theta_0, \theta_1) = \sum_{i=1}^{n} (y_{i_{real}} - y_{i_{previsto}})^2 = \sum_{i=1}^{n} (y_i - (\theta_1 * x_i + \theta_0))^2$
  - Erro Quadrático Médio:

$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} (y_{i_{real}} - y_{i_{previsto}})^2 = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\theta_1 * x_i + \theta_0))^2$$

Gradiente Descendente para Regressão Linear

- **Regressão Linear**
  - $Y = \theta_0 + \theta_1 * X$
  - $J(\theta_0, \theta_1) = \sum_{i=1}^{n} (y_{i_{real}} - y_{i_{previsto}})^2 = \sum_{i=1}^{n} (y_i - (\theta_1 * x_i + \theta_0))^2$
  - Erro Quadrático Médio:

$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} (y_{i_{real}} - y_{i_{previsto}})^2 = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\theta_1 * x_i + \theta_0))^2$$

Usar Gradiente Descendente para minimizar o Erro Quadrático Médio

- **Gradiente Descendente**
  - Repetir até convergir:
    - $temp_0 := \theta_0 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$
    - $temp_1 := \theta_1 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$
    - $\theta_0 := temp_0$
    - $\theta_1 := temp_1$

Gradiente Descendente para Regressão Linear

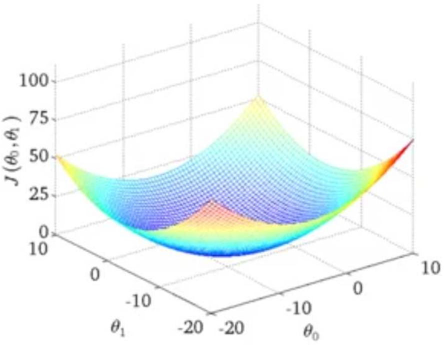
- $\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_1 * x_i + \theta_0))$

- $\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} = \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_1 * x_i + \theta_0)) * x_i$

- Repetir até convergir:
  - $temp_0 := \theta_0 - \alpha * \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_1 * x_i + \theta_0))$
  - $temp_1 := \theta_1 - \alpha * \frac{1}{n}\sum_{i=1}^{n}(y_i - (\theta_1 * x_i + \theta_0)) * x_i$
  - $\theta_0 := temp_0$
  - $\theta_1 := temp_1$

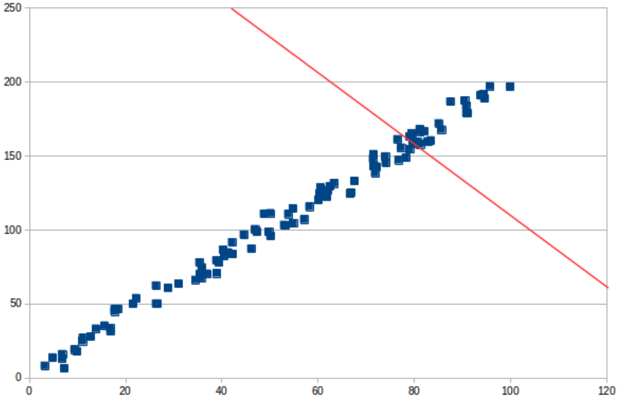
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



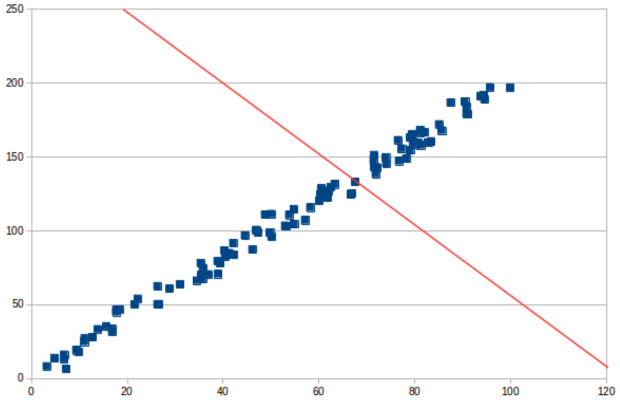
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



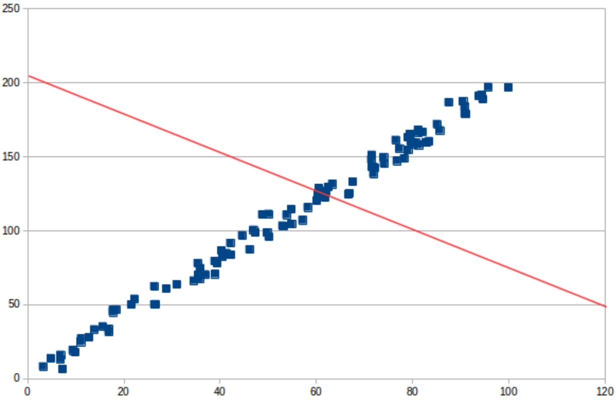
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



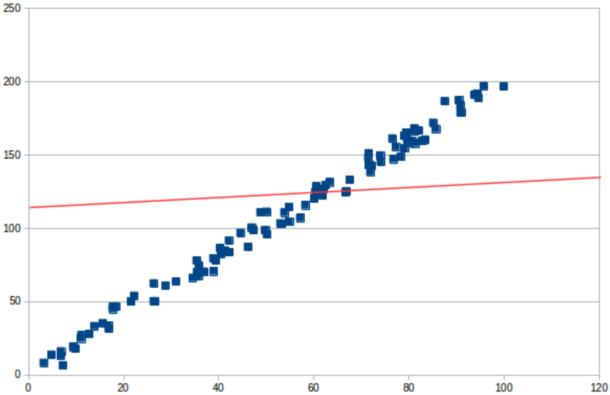
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



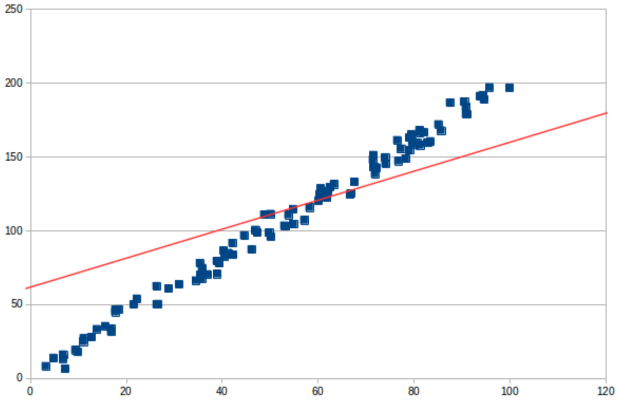
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



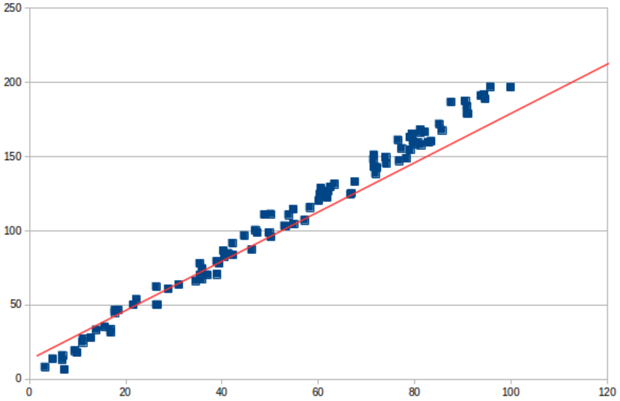
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



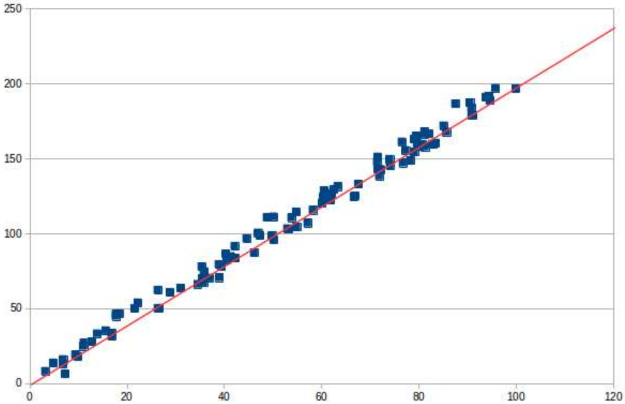
Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



Gradiente Descendente para Regressão Linear

A função de custo para a regressão linear é uma *função convexa* (possui 1 ótimo global e nenhum ótimo local):



Regressão Linear com Múltiplas Variáveis

X	Y
95.724162408	197.179636092
35.7576189281	67.5906695414
28.8168474238	60.8541328206
99.9584813087	196.907396981
66.8097483121	125.311128524
58.2156926413	115.785784589
53.8210763379	110.762772705
81.2960821704	157.98528569
80.6486970595	159.61941373
78.2528136925	149.003865539

$$y = h_\theta(x) = \theta_0 + \theta_1 x$$

Regressão Linear com Múltiplas Variáveis

X_1	X_2	X_3	**Y**
95.724162408	5	45.08	197.179636092
35.7576189281	3	34.32	67.5906695414
28.8168474238	3	11.12	60.8541328206
99.9584813087	2	32.32	196.907396981
66.8097483121	7	67.76	125.311128524
58.2156926413	1	21.21	115.785784589
53.8210763379	2	44.76	110.762772705
81.2960821704	0	55.32	157.98528569
80.6486970595	6	2.21	159.61941373
78.2528136925	4	66.54	149.003865539

- n: número de features. Na tabela, $n = 3$.
- $x^{(i)}$: entrada do i-ésimo exemplo de treinamento. Exemplo: $x^{(2)} = [35.7576189281, 3, 34.32]$
- $x_j^{(i)}$: valor da j-ésima feature no i-ésimo exemplo de treinamento. Exemplo: $x_3^{(2)} = 34.32$

Regressão Linear com Múltiplas Variáveis

X_1	X_2	X_3	**Y**
95.724162408	5	45.08	197.179636092
35.7576189281	3	34.32	67.5906695414
28.8168474238	3	11.12	60.8541328206
99.9584813087	2	32.32	196.907396981
66.8097483121	7	67.76	125.311128524
58.2156926413	1	21.21	115.785784589
53.8210763379	2	44.76	110.762772705
81.2960821704	0	55.32	157.98528569
80.6486970595	6	2.21	159.61941373
78.2528136925	4	66.54	149.003865539

Antes:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

E agora?

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

Generalizando

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Regressão Linear com Múltiplas Variáveis

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Considere $x_0 = 1 (x_0^{(i)} = 1)$. Logo: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Fazendo: $x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ e $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdots \\ \theta_n \end{bmatrix}$

temos: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = [\theta_0, \theta_1, \cdots, \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$

Regressão Linear com Múltiplas Variáveis: $h_\theta(x) = \theta^T x$

Regressão Linear com Múltiplas Variáveis - Gradiente Descendente

- Hipótese

$$h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

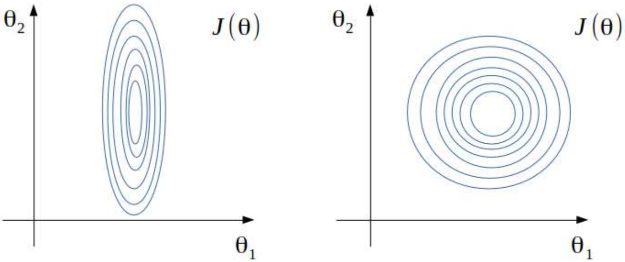
- Parâmetros: $\theta : \theta_0, \theta_1, \cdots, \theta_n$
- Função de Custo: $J(\theta) = J(\theta_0, \theta_1, \cdots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Gradiente Descendente

- Repetir até convergir:
  - $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \cdots, \theta_n) = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$
- Atualização simultânea.

Normalização

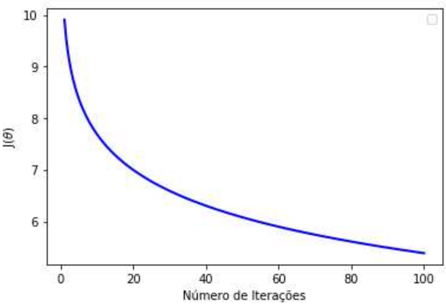
- Normalizando as features: diferentes features fiquem na mesma escala.
- x_1 variando de 0-2000 e x_2 variando de 1-5.
- $x_1 = \frac{x_1}{2000}$ e $x_2 = \frac{x_2}{5}$
- Ter uma convergência mais rápida.



- **Mean Normalization**: $x_i = \frac{x_i - \mu_i}{total}$ $(x_1 = \frac{x_1 - 1000}{2000}$ e $x_2 = \frac{x_2 - 2}{5})$
- Toda feature: aproximadamente $-1 \leq x_i \leq 1$

Gradiente Descendente

- $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$
- Como saber se o gradiente descendente está funcionando corretamente e como escolher α?



- $J(\theta)$ deve decrescer a cada iteração.
- Se $J(\theta)$ estiver crescendo, utilize um α menor. Isso pode retardar a convergência. $(\alpha = \cdots, 0.001, \cdots, 0.01, \cdots, 0.1, \cdots)$

Inteligência Artificial
Aula 10 - vídeo 2 - Regressão Linear e Logística

João C. P. da Silva

Dept. Ciência da Computação - UFRJ

28 de outubro de 2020