

Regressão Linear

Prof. Dr. Leandro Balby Marinho



Aprendizagem de Máquina

Roteiro

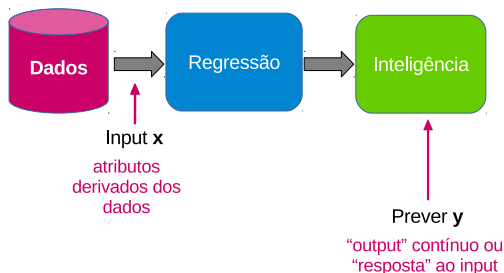
1. Introdução

2. Regressão Linear Simples

3. Aprendizado de Parâmetros

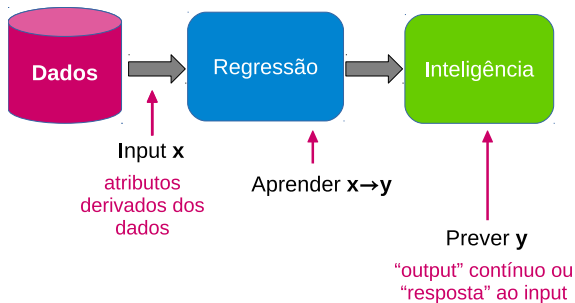
Regressão

De atributos para previsão.



Regressão

De atributos para previsão.



Salário depois de formado

- ▶ De quanto será o seu salário depois de formado? ($y = \text{R\$}$)
- ▶ Depende de $x =$ anos de estudo, desempenho geral, desempenho em disciplinas específicas, participação em projetos, fluência em inglês, ...

Previsão de preços de ações

- ▶ Qual será o preço de determinada ação amanhã? (y).
- ▶ Depende de x = histórico de preço recente da ação, notícias recentes, commodities relacionadas,....

Popularidade de Tweet

- ▶ Quantas pessoas vão retuitar o meu tweet? (y).
- ▶ Depende de $x = \#$ seguidores, atributos do texto tuitado, popularidade da hashtag, $\#$ retweets passados,

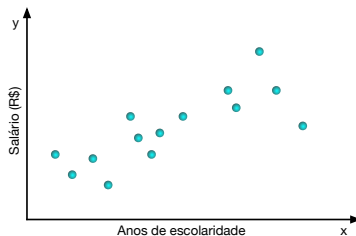
Predição de Salário

Anos de Escolaridade	Salário Anual (em milhares de R\$)
8	26
8	21
10	26
11	36
\vdots	\vdots

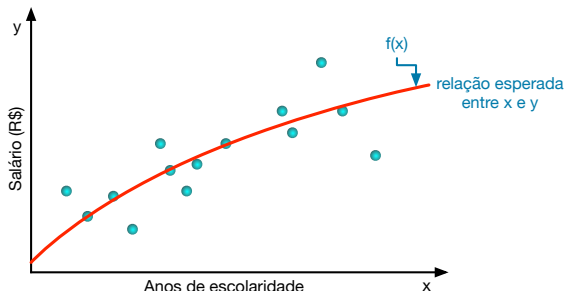
Dado que eu tenho x anos de escolaridade, qual será meu salário?

Componentes da Aprendizagem

- ▶ Entrada: x
- ▶ Saída: y
- ▶ Função alvo: $f : \mathcal{X} \rightarrow \mathcal{Y}$
- ▶ Dados de Treino: $\mathcal{D}^{\text{train}} := \{(x_1, y_1), \dots, (x_N, y_N)\}$

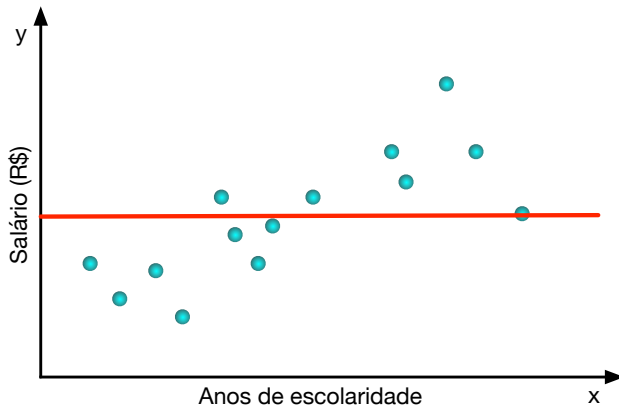


Modelo: Como assumimos que o mundo funciona

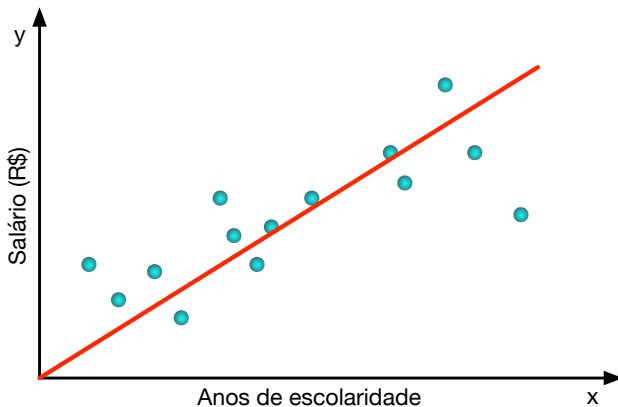


Modelo de Regressão: $y_i = f(x_i) + \epsilon_i$, tal que $E[\epsilon] = 0$

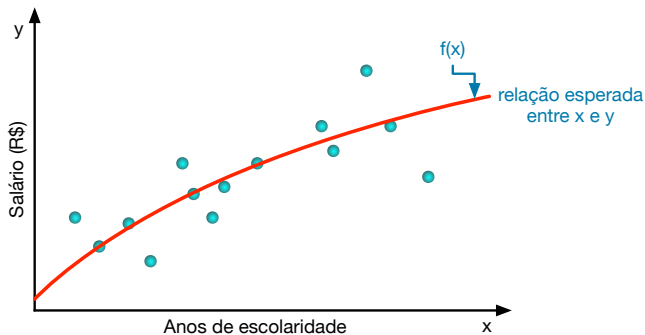
Tarefa 1: Que modelo usar?



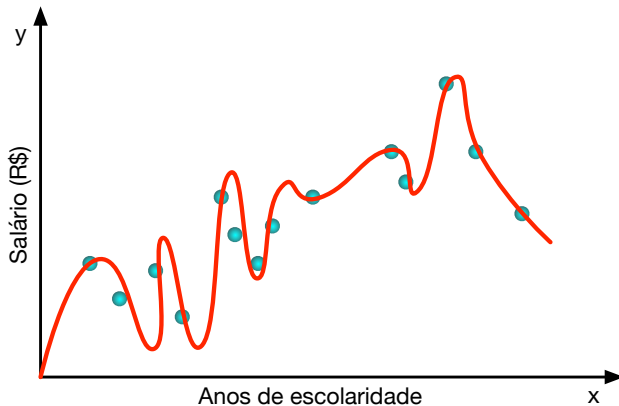
Tarefa 1: Que modelo usar?



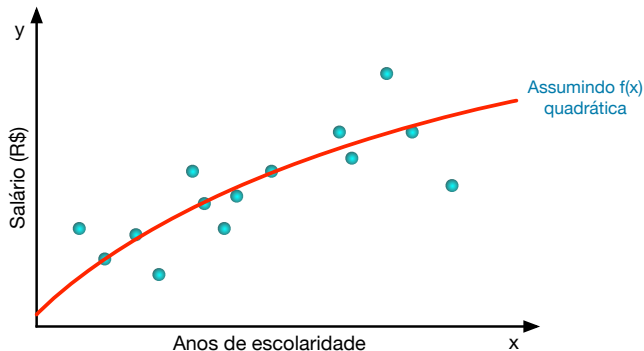
Tarefa 1: Que modelo usar?



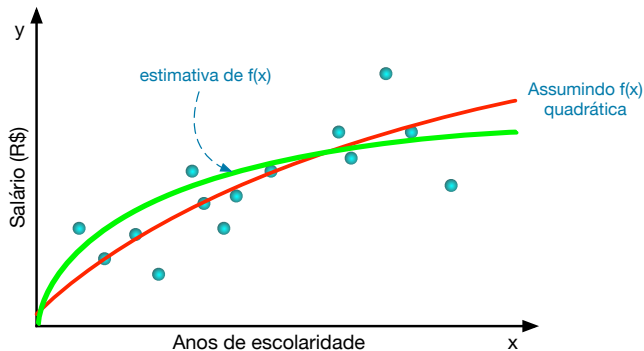
Tarefa 1: Que modelo usar?



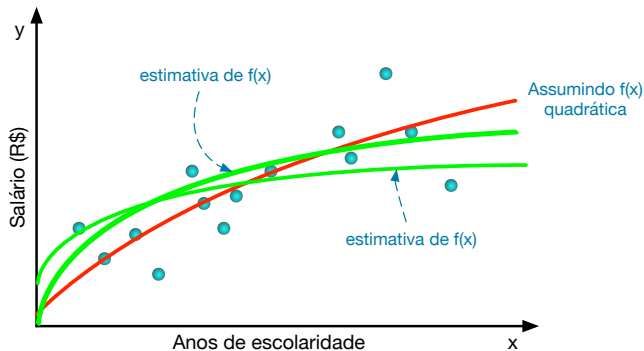
Tarefa 2: Dado $f(x)$, como estimar $\hat{f}(x)$ dos dados?



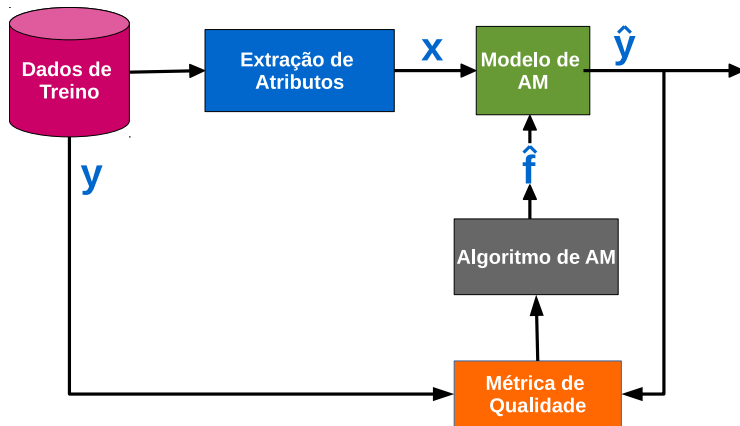
Tarefa 2: Dado $f(x)$, como estimar $\hat{f}(x)$ dos dados?



Tarefa 2: Dado $f(x)$, como estimar $\hat{f}(x)$ dos dados?



Regressão Workflow



Roteiro

1. Introdução

2. Regressão Linear Simples

3. Aprendizado de Parâmetros

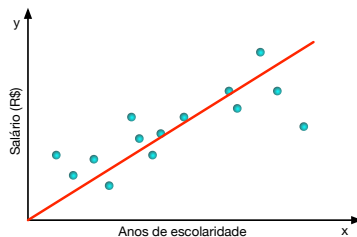
Regressão Linear Simples

Assume-se que a relação entre a variável de entrada e saída é **linear**:

$$f(x) = w_0 + w_1x$$

onde w_0 e w_1 são chamados de parâmetros do modelo. Cada observação é então definida por

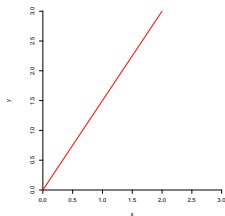
$$y_i = w_0 + w_1x_i + \epsilon_i$$



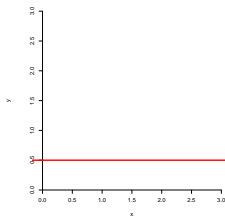
Parâmetros do Modelo

$w_0 \dots$ Coeficiente linear

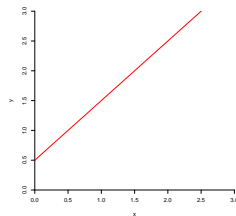
$w_1 \dots$ Coeficiente angular



$$w_0 = 0, w_1 = 1.5$$



$$w_0 = 0.5, w_1 = 0$$

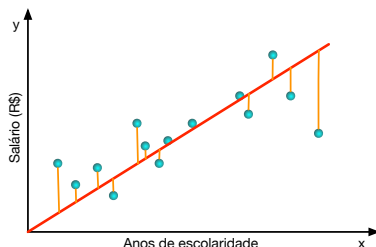


$$w_0 = 0.5, w_1 = 1.5$$

Custo de uma única linha de regressão

Custo: Soma dos erros quadrados

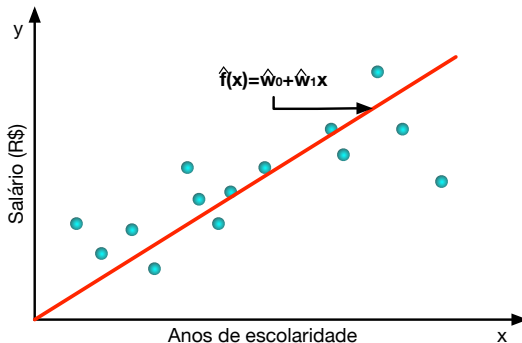
$$\text{RSS}(w_0, w_1) = \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])^2$$



Para diferentes escolhas de w_0 e w_1 tem-se diferentes RSS.

Modelo vs linha ajustada

- ▶ Modelo de regressão linear: $y_i = w_0 + w_1 + \epsilon_i$
- ▶ Parâmetros estimados: \hat{w}_0, \hat{w}_1

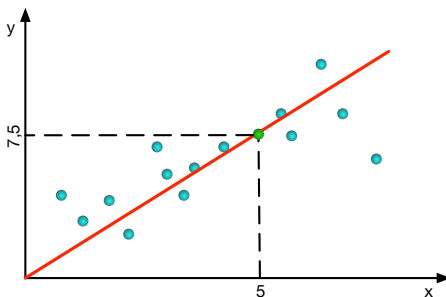


Usando o modelo aprendido

- ▶ Por exemplo, para $x = 5$:

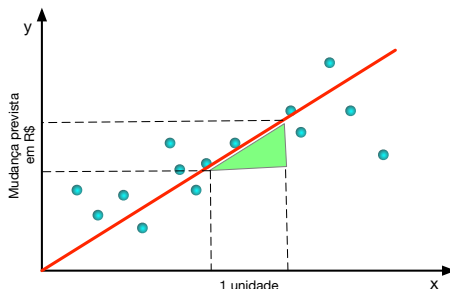
$$\hat{y} = \hat{f}(5) = \hat{w}_0 + 5\hat{w}_1$$

- ▶ Assumindo por exemplo: $\hat{w}_0 = 0, \hat{w}_1 = 1.5$
- ▶ $\hat{y} = 7,5$



Interpretando a Linha de Regressão

- ▶ $\hat{y} = \hat{w}_0 + \hat{w}_1 x$
- ▶ $w_0 \dots$ valor de \hat{y} quando $\hat{w}_1 = 0$
- ▶ $w_1 \dots$ mudança prevista em \hat{y} por mudança de uma unidade em x .



Roteiro

1. Introdução
2. Regressão Linear Simples
3. Aprendizado de Parâmetros

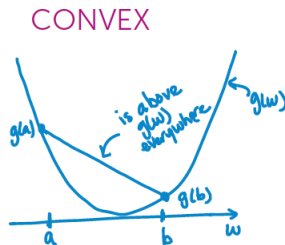
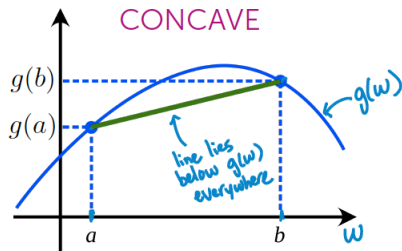
Regressão como um Problema de Otimização

- ▶ **Ideia: Escolha w_0, w_1 tal que $\hat{y} \approx y$ nos dados de treino.**
- ▶ Especificamente, escolha w_0, w_1 tal que o RSS seja mínimo:

$$\min_{w_0, w_1} \text{RSS}(w_0, w_1)$$

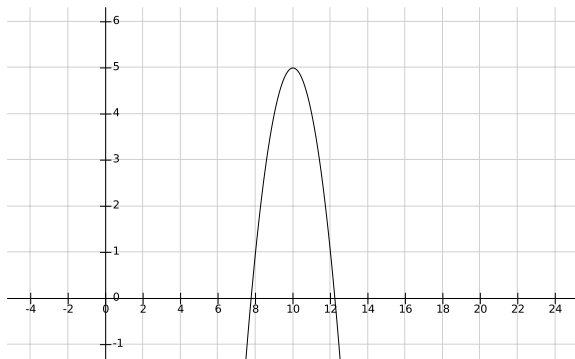
- ▶ Esse método também é chamado de mínimos quadrados ou *Ordinary Least Squares (OLS)*

Funções Côncavas e Convexas



Máximos e Mínimos em uma Dimensão

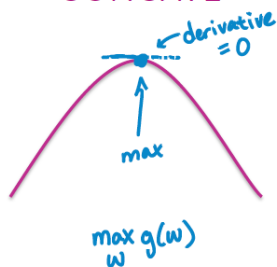
Qual o valor de w que maximiza a função $g(w) = 5 - (w - 10)^2$?



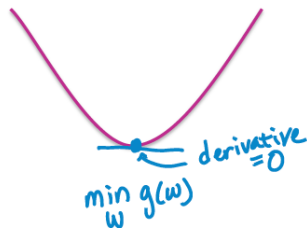
Calcule a derivada e iguale a zero (por que?)

Achando Máximos e Mínimos de Forma Analítica

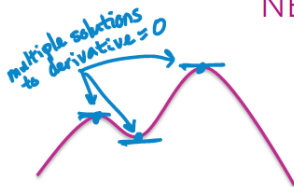
CONCAVE



CONVEX



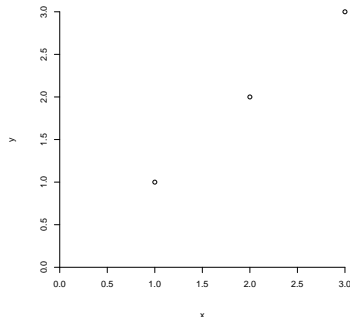
NEITHER



Formato da Função de Erro

Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.

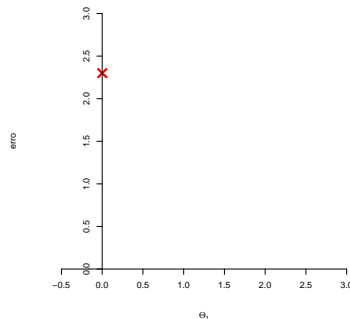
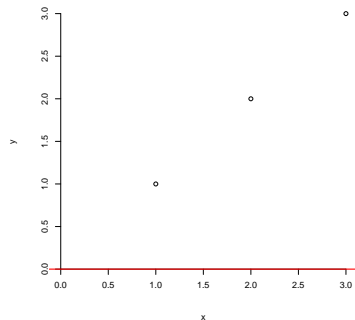


Formato da Função de Erro

Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.

$\text{RSS}(w_1 = 0)$

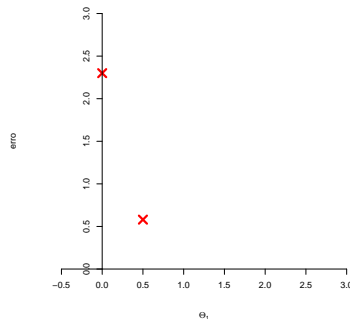
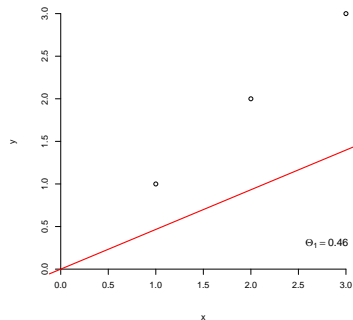


Formato da Função de Erro

Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.

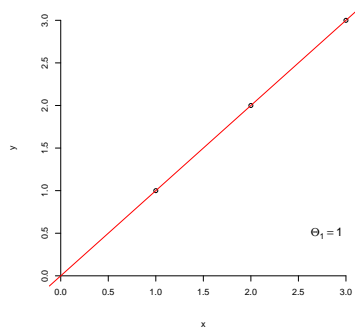
$\text{RSS}(w_1 = 0.5)$



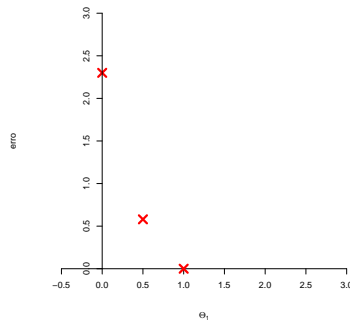
Formato da Função de Erro

Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.



$\text{RSS}(w_1 = 1)$

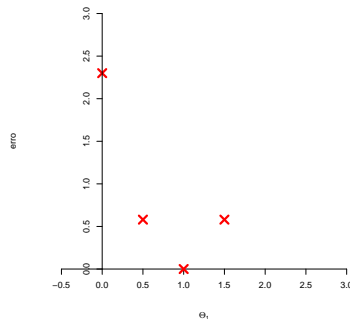
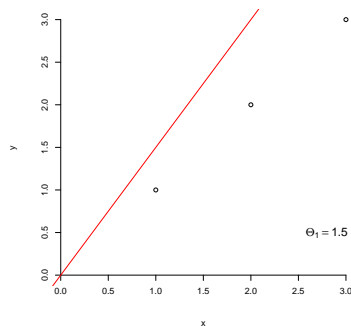


Formato da Função de Erro

Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.

$\text{RSS}(w_1 = 1.5)$

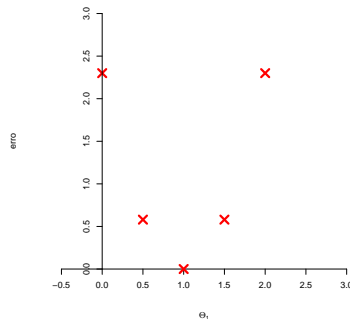
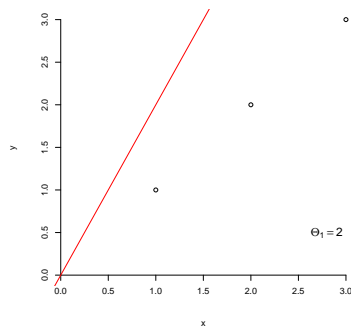


Formato da Função de Erro

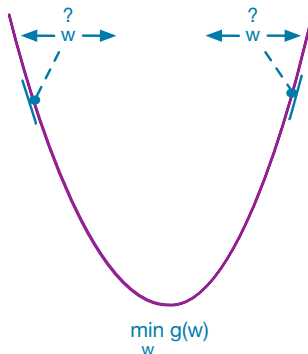
Considere $\mathcal{D}^{\text{train}} = \{(1, 1), (2, 2), (3, 3)\}$

Mantendo $w_0 = 0$ fixo.

$\text{RSS}(w_1 = 2.0)$

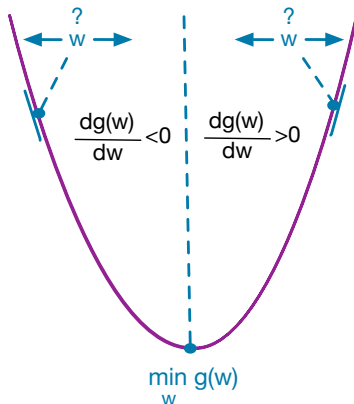


Mínimo via Hill Descent



- ▶ Quando a derivada é positiva queremos diminuir w .
- ▶ Quando negativa queremos aumentar w .

Mínimo via Hill Descent



Hill-Descent

- 1 **while** not converged
- 2 $w^{(t+1)} = w^t - \alpha \frac{d}{dw} g(w^t)$

Derivadas Parciais

Para uma função multivariada, como $f(x, y) = x^2y$, calcular derivadas parciais se resume a:

$$\frac{\partial f}{\partial x} = \underbrace{\frac{\partial}{\partial x} x^2 y}_{\text{trate } y \text{ como constante}} = 2xy$$

$$\frac{\partial f}{\partial y} = \underbrace{\frac{\partial}{\partial y} x^2 y}_{\text{trate } x \text{ como constante}} = x^2 \cdot 1$$

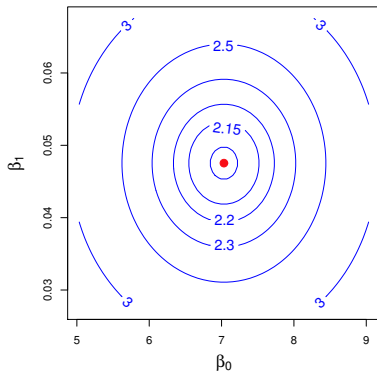
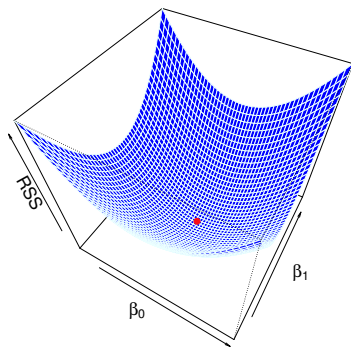
Gradiente

O gradiente de uma função multivariada $f(x, y, \dots)$, denotada por ∇f , empacota todas suas derivadas parciais em um vetor:

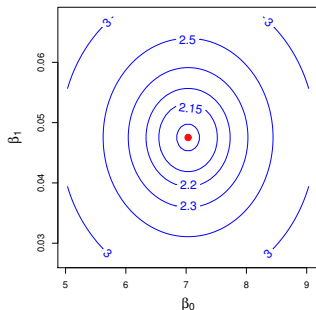
$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \vdots \end{pmatrix}$$

O gradiente aponta para a direção onde a função está mudando mais rapidamente.

Formato da Função de Erro em duas Dimensões



Gradiente Descendente



Gradient-Descent

- 1 **while** not converged
- 2 $w^{(t+1)} = w^t - \alpha \nabla g(w^t)$

Note que agora w e $\nabla g(w)$ são vetores.

Calculando o gradiente de RSS

Lembrando que:
$$\text{RSS}(w_0, w_1) = \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])^2$$

Derivada em relação a w_0 :

$$\frac{\partial}{\partial w_0} \text{RSS}(w_0, w_1) = \sum_{i=1}^N \frac{\partial}{\partial w_0} (y_i - [w_0 + w_1 x_i])^2$$

$$\frac{\partial}{\partial w_1} \text{RSS}(w_0, w_1) = \sum_{i=1}^N \frac{\partial}{\partial w_1} (y_i - [w_0 + w_1 x_i])^2$$

Note que a derivada da soma é a soma das derivadas.

Derivada em relação a w_0 :

$$\text{Lembrando que: } \text{RSS}(w_0, w_1) = \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])^2$$

$$\begin{aligned} \frac{\partial}{\partial w_0} \text{RSS}(w_0, w_1) &= \sum_{i=1}^N 2(y_i - [w_0 + w_1 x_i])(-1) \\ &= -2 \sum_{i=1}^N (y_i - [w_0 + w_1 x_i]) \end{aligned}$$

Derivada em relação a w_1

Lembrando que:
$$\text{RSS}(w_0, w_1) = \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])^2$$

$$\begin{aligned}\frac{\partial}{\partial w_1} \text{RSS}(w_0, w_1) &= \sum_{i=1}^N 2(y_i - [w_0 + w_1 x_i])(-x_i) \\ &= -2 \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])x_i\end{aligned}$$

Gradiente de RSS

$$\nabla \text{RSS}(w_0, w_1) = \begin{pmatrix} -2 \sum_{i=1}^N (y_i - [w_0 + w_1 x_i]) \\ -2 \sum_{i=1}^N (y_i - [w_0 + w_1 x_i]) x_i \end{pmatrix}$$

Estimativa dos Coeficientes

Podemos achar os parâmetros ótimos de forma fechada, igualando suas derivadas a 0.

$$\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}$$

$$\hat{w}_1 = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

Também chamadas de equações normais.

Prova para w_0

$$\frac{\partial}{\partial w_0} \text{RSS}(w_0, w_1) = -2 \sum_{i=1}^N (y_i - [w_0 + w_1 x_i]) = 0$$

$$Nw_0 = \sum_{i=1}^N y_i - w_1 \sum_{i=1}^N x_i$$

$$w_0 = \frac{1}{N} \sum_{i=1}^N y_i - \frac{w_1}{N} \sum_{i=1}^N x_i$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

Algoritmo Regressão Simples

RegSimples($\mathcal{D}^{\text{train}}$)

```
1  tmpx = 0
2  tmpy = 0
3  for  $i = 1$  to  $N$ 
4      tmpx = tmpx +  $x_i$ 
5      tmpy = tmpy +  $y_i$ 
6   $\bar{x} = \text{tmp}_x / N$ 
7   $\bar{y} = \text{tmp}_y / N$ 
8   $a = 0$ 
9   $b = 0$ 
10 for  $i = 1$  to  $n$ 
11      $a = a + (x_i - \bar{x})(y_i - \bar{y})$ 
12      $b = b + (x_i - \bar{x})^2$ 
13  $w_1 = a / b$ 
14  $w_0 = \bar{y} - w_1 \bar{x}$ 
15 return ( $w_0, w_1$ )
```

Gradiente Descendente




- ▶ Comece com algum valor para w_0, w_1 .
- ▶ Atualize w_0, w_1 iterativamente, **reduzindo** $RSS(w_0, w_1)$, até atingir o mínimo.
- ▶ **Ideia: Atualize w_0, w_1 proporcionalmente as derivadas parciais (gradiente) da função de erro em relação a w_0, w_1 .**

Algoritmo Gradiente Descendente

GradientDescent(α, ϵ)

```
1  initialize  $w_0, w_1$ 
2  while  $\|\nabla \text{RSS}(w_0, w_1)\| \geq \epsilon$ 
3       $\text{tmp}_0 = w_0 + 2\alpha \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])$ 
4       $\text{tmp}_1 = w_1 + 2\alpha \sum_{i=1}^N (y_i - [w_0 + w_1 x_i])(x_i)$ 
5       $w_0 = \text{tmp}_0$ 
6       $w_1 = \text{tmp}_1$ 
7  return ( $w_0, w_1$ )
```

Referências

-  Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning with Applications in R. Springer, 2013.
-  Yaser S. Abu-Mostafa, Malik Magdon-Ismail. Learning from Data. AMLBook, 2012.
-  Emily Fox and Carlos Guestrin. Machine Learning Specialization. Curso online disponível em <https://www.coursera.org/specializations/machine-learning>. Último acesso: 31/08/2017.