Regressão Linear

Prof. Danilo Silva

EEL7514/EEL7513 - Tópico Avançado em Processamento de Sinais:
Introdução ao Aprendizado de Máquina

EEL / CTC / UFSC

Tópicos

- Regressão: Conceitos gerais
- Regressão linear
- Função custo
- Otimização analítica (equações normais)
- Regressão linear com funções de base
- Método do gradiente

Regressão: Conceitos Gerais

Regressão

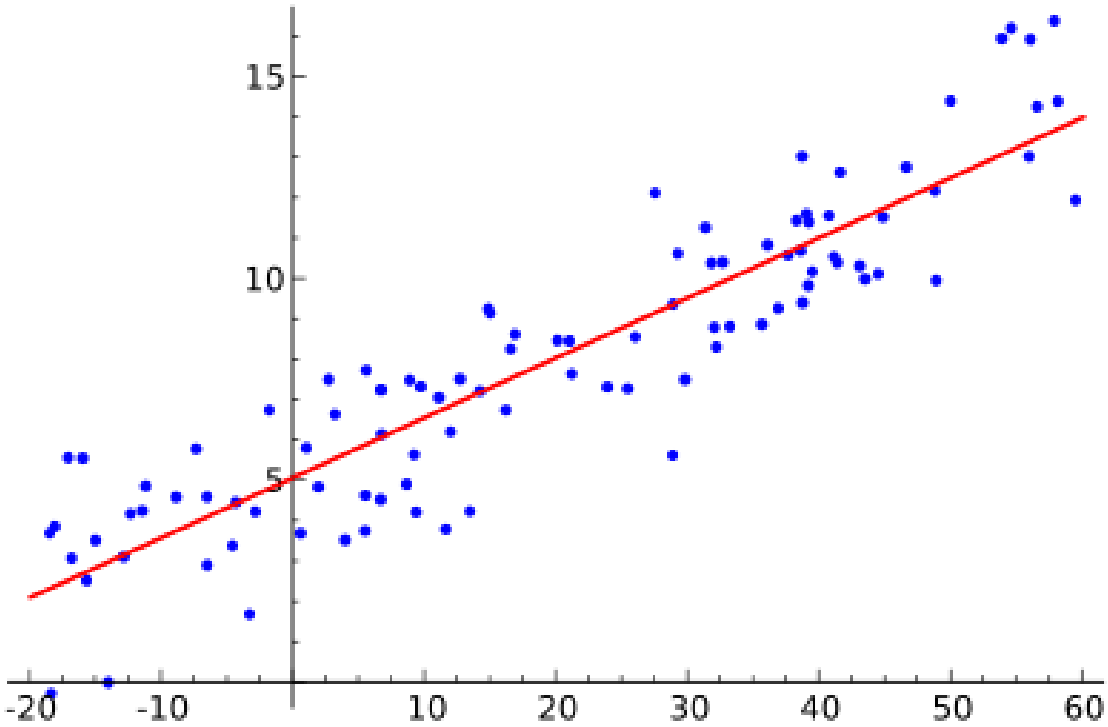
	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

- Variáveis aleatórias:
  - $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ é o vetor de entrada ou vetor de atributos
  - $y \in \mathbb{R}$ é a variável de saída ou valor-alvo ou rótulo correspondente a $\mathbf{x}$
- A distribuição $p(\mathbf{x}, y)$ é desconhecida, mas conhecemos m amostras $(\mathbf{x}^{(i)}, y^{(i)})$ dessa distribuição chamadas de exemplos de treinamento
- Conjunto de treinamento: $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$

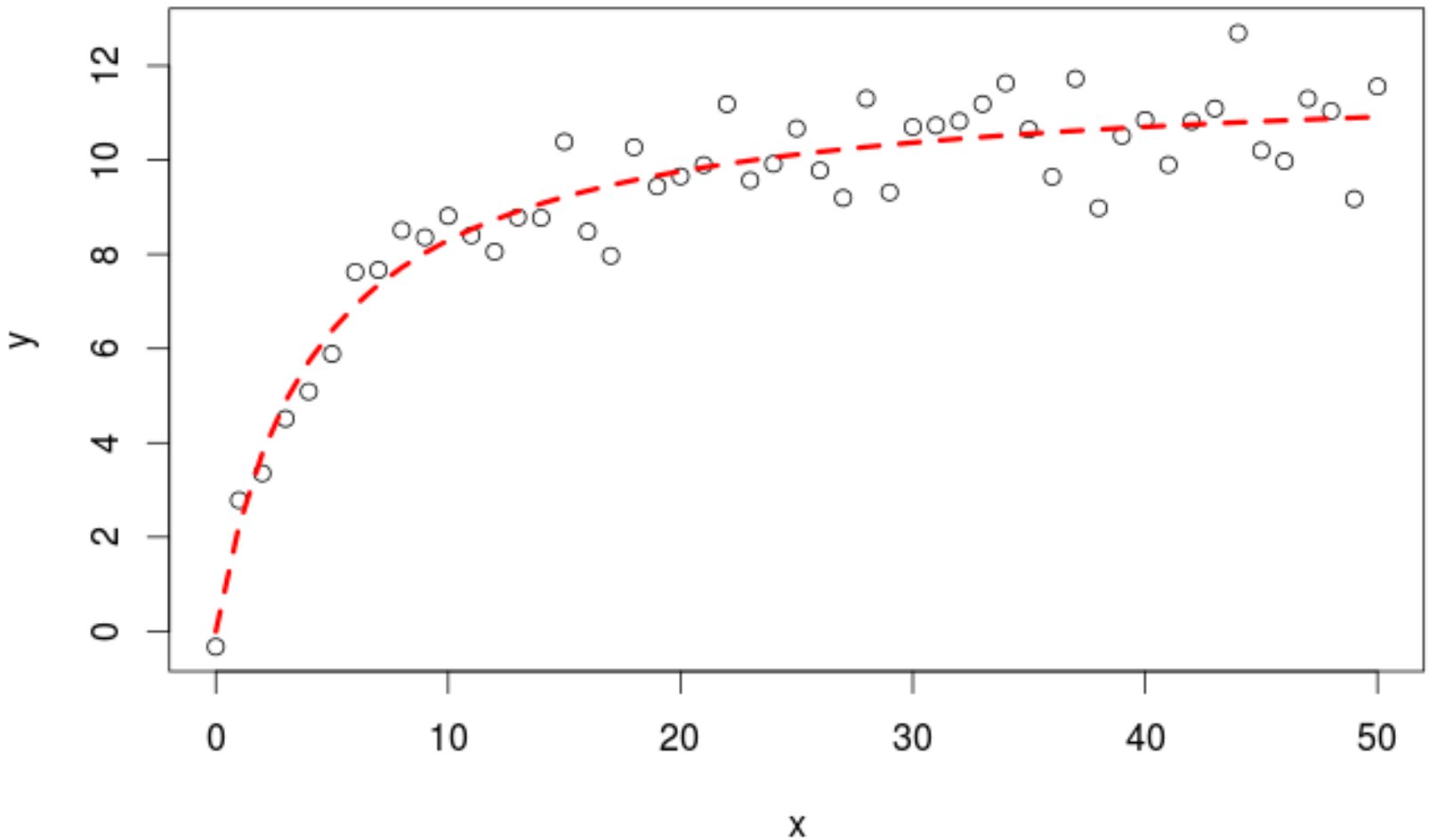
Regressão

- Um modelo de aprendizado determina uma função hipótese $g : \mathbb{R}^n \to \mathbb{R}$ que realiza a predição $y = g(\mathbf{x})$
  - A função g é escolhida dentre um espaço de hipóteses $\mathcal{G}$
- É intuitivo exigir $g(\mathbf{x}) \approx y$ para $(\mathbf{x}, y) \in \mathcal{D}$, mas o que realmente importa é qualidade da predição para novos exemplos $(\mathbf{x}, y) \notin \mathcal{D}$ fora do conjunto de treinamento

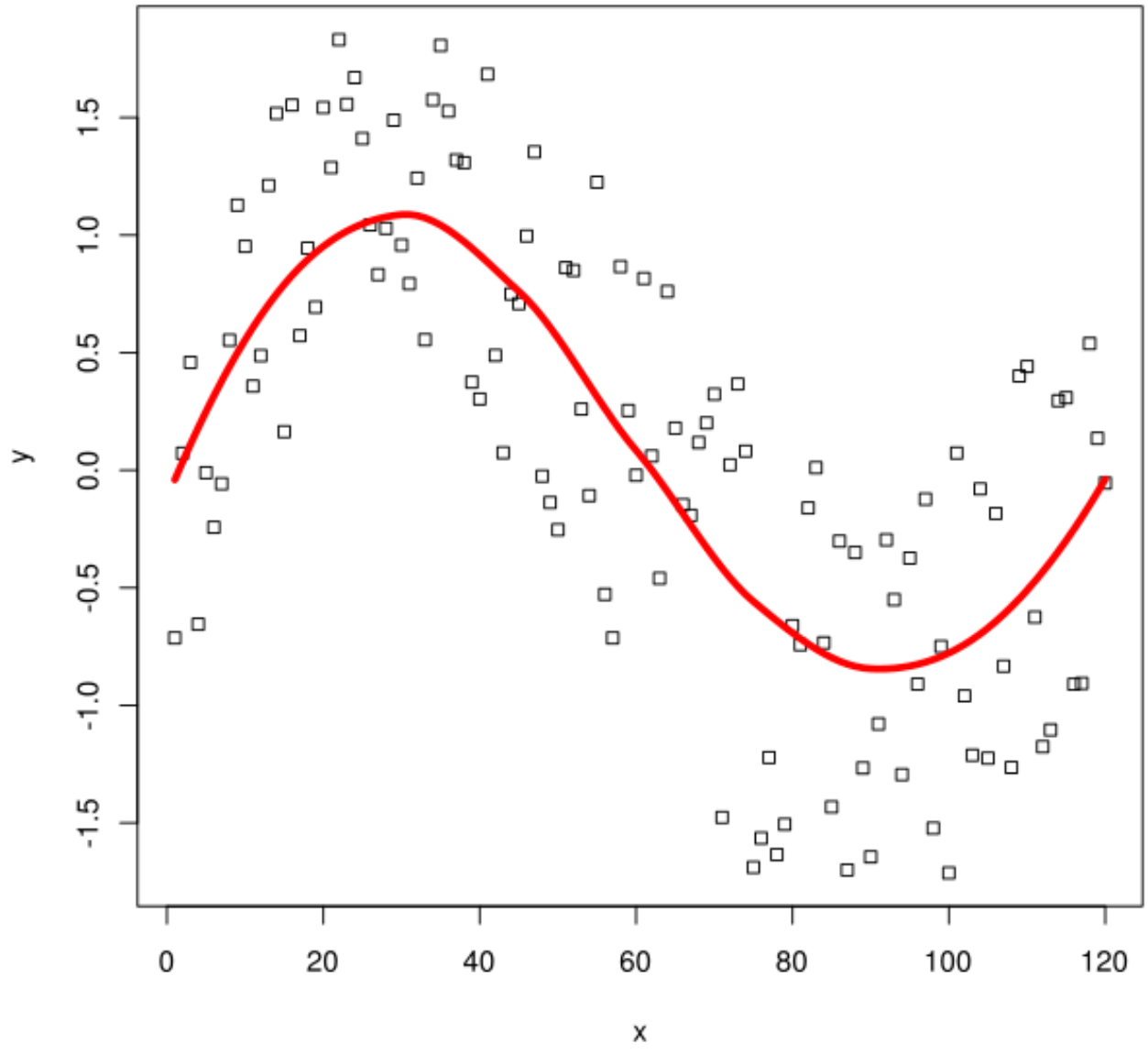
Exemplos



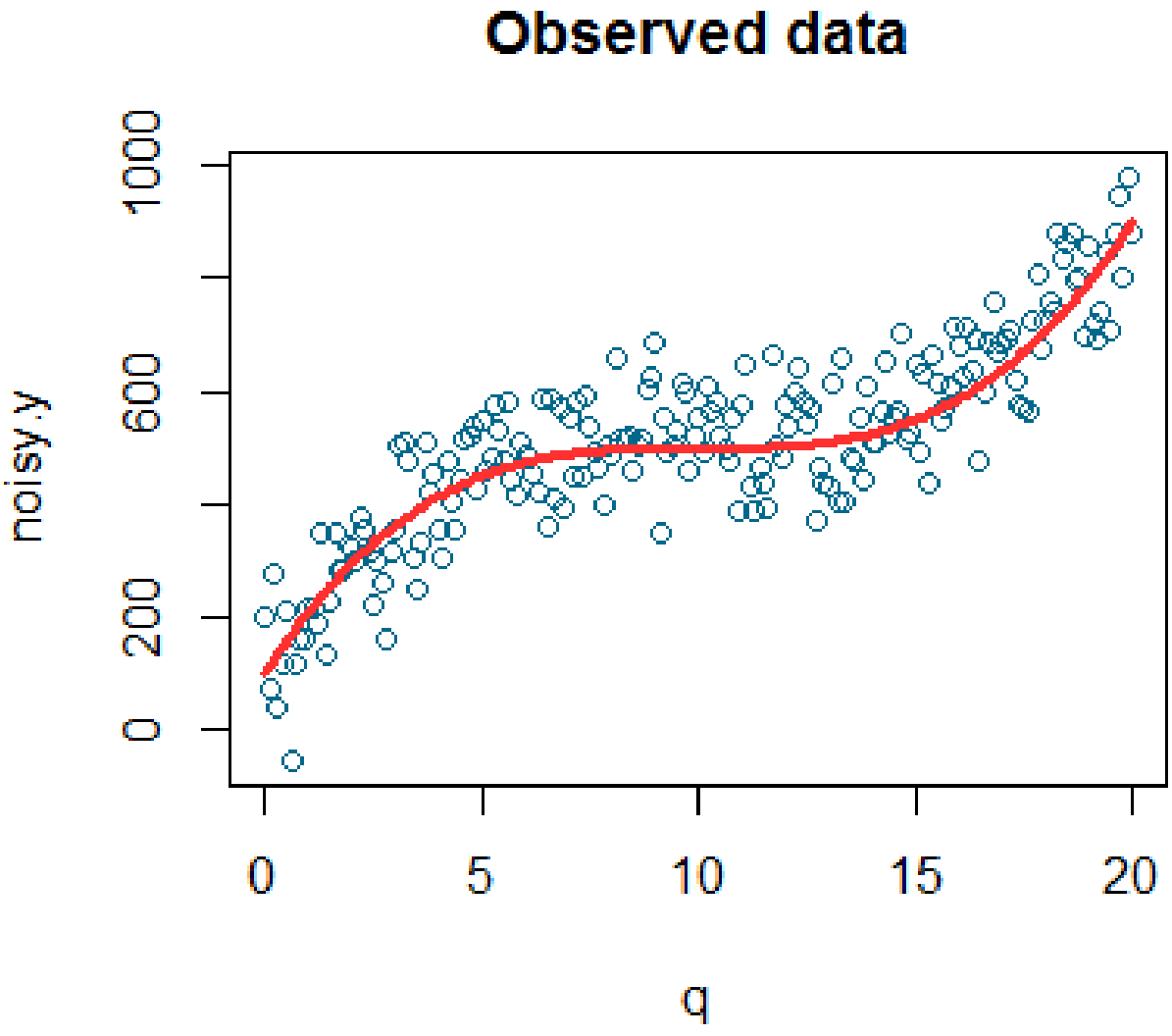
Exemplos



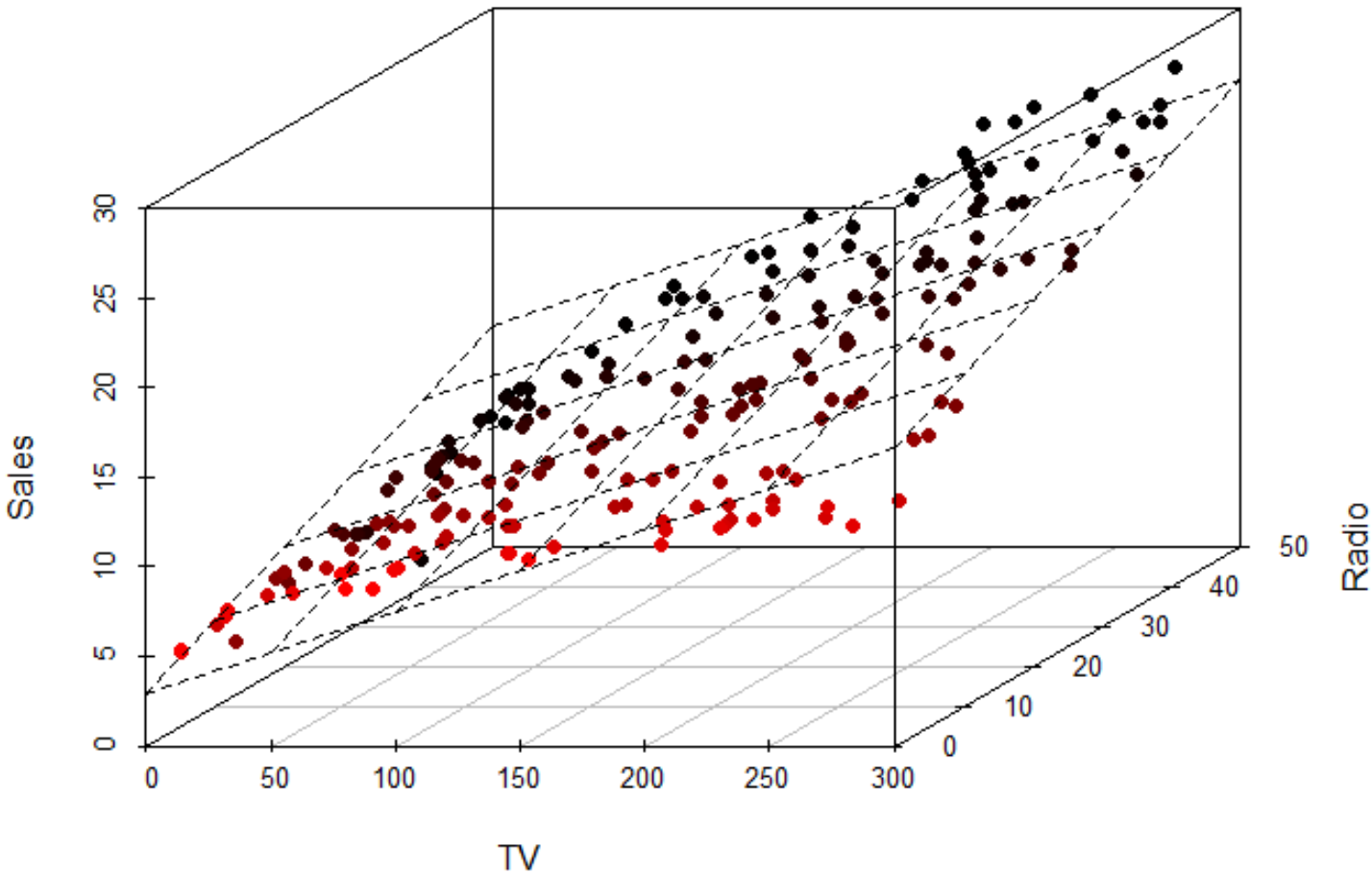
Exemplos



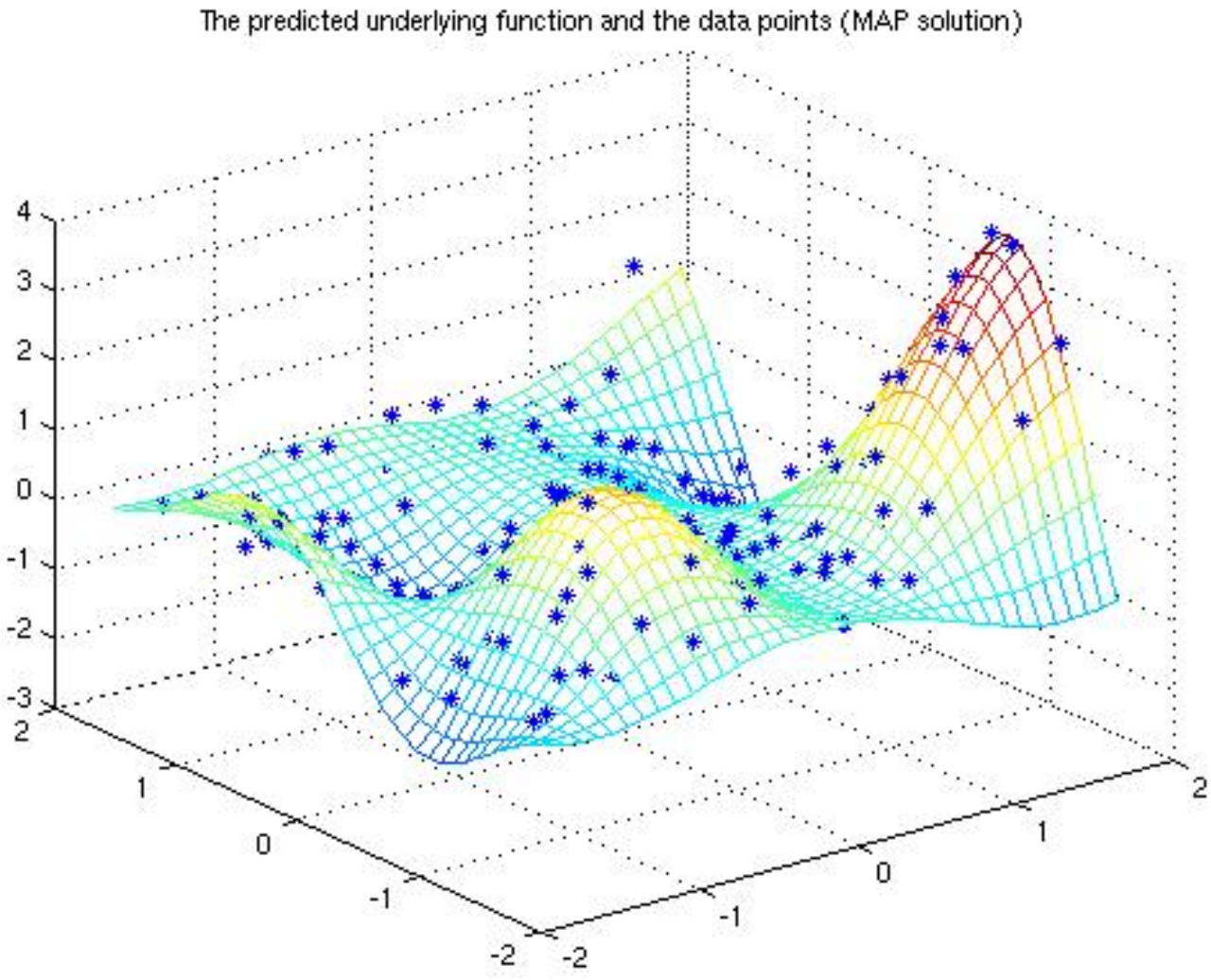
Exemplos



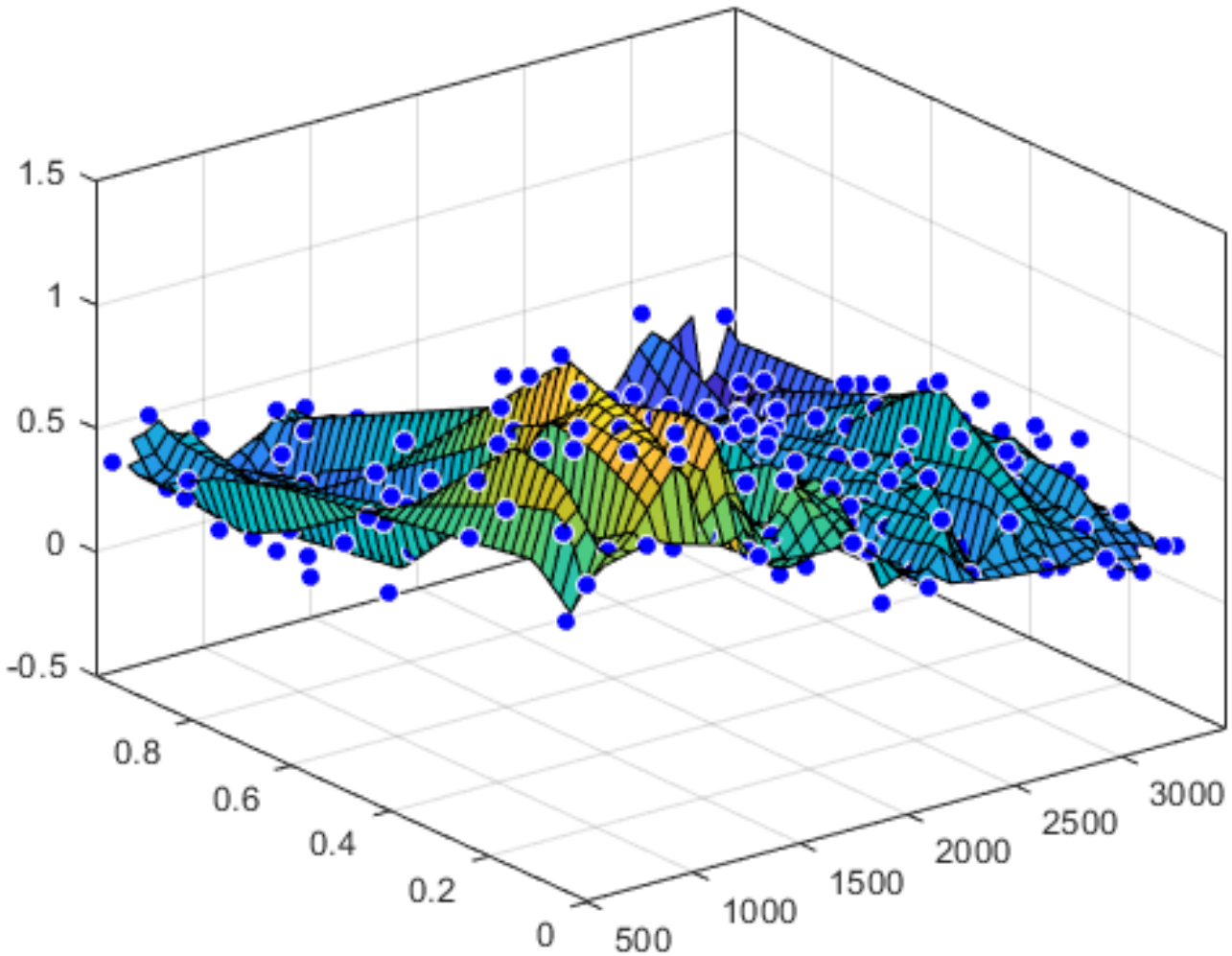
Exemplos


Sales
0
5
10
15
20
25
30
TV
0
50
100
150
200
250
300
Radio
0
10
20
30
40
50

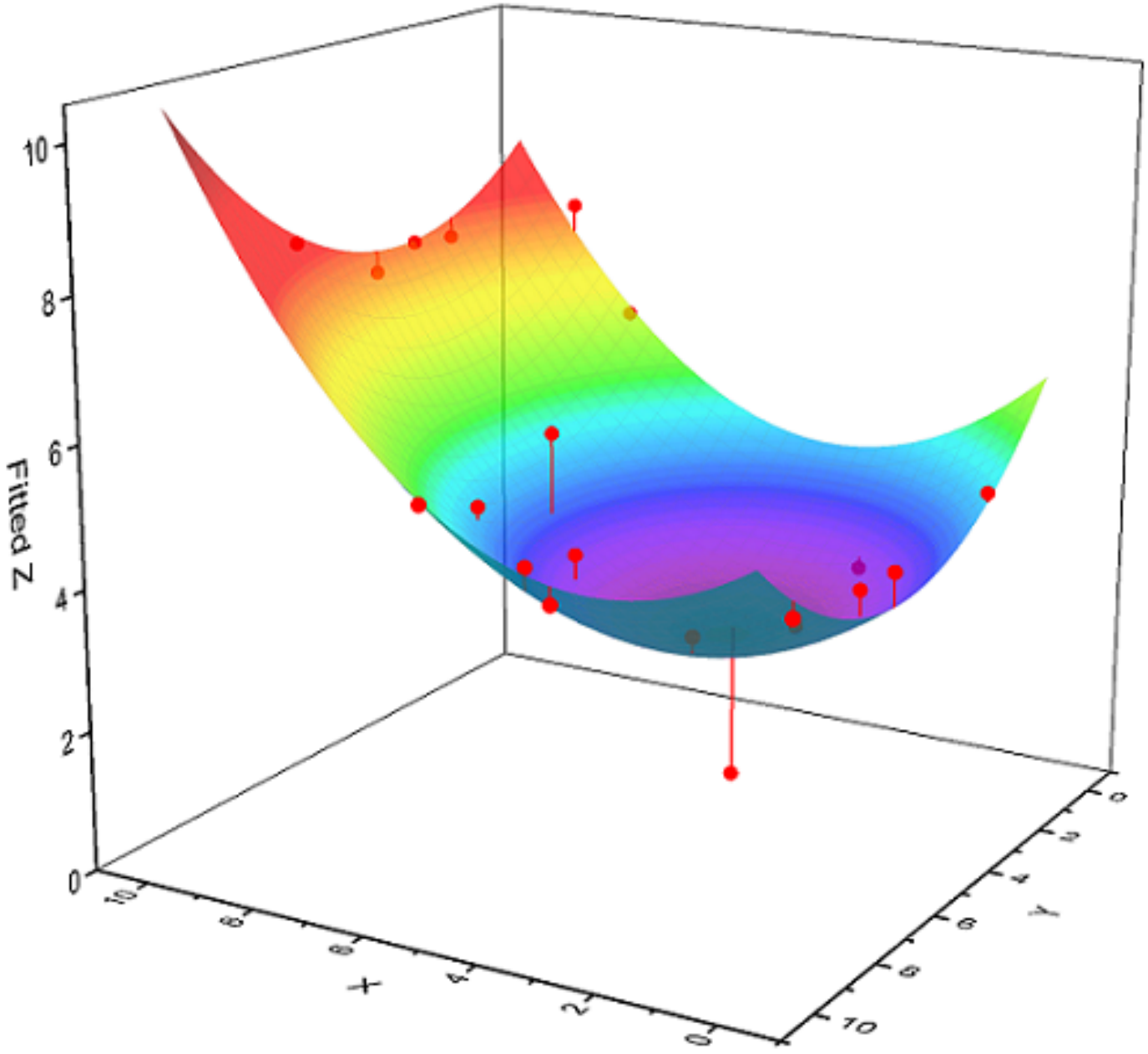
Exemplos


The predicted underlying function and the data points (MAP solution)
4
3
2
1
0
-1
-2
-3
2
1
0
-1
-2
-2
-1
0
1
2

Exemplos



Exemplos



Função Perda

- Para avaliar um modelo é preciso definir uma métrica de desempenho, isto é, uma função erro ou função perda (*loss function*) $L(\hat{y}, y)$
  - Indica quão "ruim" é a predição $\hat{y}$ quando o valor-alvo correto é y
- Exemplos:
  - Erro absoluto:
    $$L(\hat{y}, y) = |\hat{y} - y|$$
  - Erro quadrático:
    $$L(\hat{y}, y) = (\hat{y} - y)^2$$
    ou
    $$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$
    - Vantagem: Mais conveniente matematicamente, diferenciável em todos os pontos
    - Desvantagem: Mais sensível a *outliers*

Erro Médio de um Modelo

- Para medir o desempenho de um modelo no conjunto de treinamento, é usual calcular o erro médio (média aritmética) sobre todo o conjunto:

$$J(g) = \frac{1}{m} \sum_{(\mathbf{x},y)\in\mathcal{D}} L(g(\mathbf{x}), y) = \frac{1}{m} \sum_{i=1}^{m} L(g(\mathbf{x}^{(i)}), y^{(i)})$$

- Para avaliar o poder preditivo de um modelo (generalização), deve-se medir o desempenho sobre um conjunto de teste $\mathcal{D}_{\text{test}}$ gerado de forma independente do conjunto de treinamento
  - Nesse caso denotamos o erro médio por $J_{\text{test}}(g)$:

$$J_{\text{test}}(g) = \frac{1}{m_{\text{test}}} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{test}}} L(g(\mathbf{x}), y) = \frac{1}{m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} L(g(\mathbf{x}_{\text{test}}^{(i)}), y_{\text{test}}^{(i)})$$

Treinamento

- O erro no conjunto de treinamento é usado para determinar os parâmetros do modelo, isto é, para selecionar a hipótese $g \in \mathcal{G}$ (dentre um espaço de hipóteses pré-definido) que melhor se ajusta aos dados de treinamento
  - Treinamento também é chamado de ajuste (*fit*)
- A função $J(g)$ é usada como função objetivo (ou função custo) de um algoritmo de otimização:

$$\min_{g \in \mathcal{G}} J(g)$$

Regressão Linear

Modelo Linear para Regressão

- Função-hipótese:

$$\hat{y} = g(\mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_n x_n$$

- Parâmetros do modelo: $w_0, w_1, \ldots, w_n$
- Se $n = 1$, temos uma reta

$$\hat{y} = g(x) = w_0 + w_1 x = b + wx$$

onde $w = w_1$ é o coeficiente de inclinação e $b = w_0$ é a intersecção com o eixo y (*intercept term*), também chamado de *bias*

Notação Vetorial

- Em notação vetorial, temos

$$\hat{y} = g(\mathbf{x}) = w_0 + \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = b + \mathbf{w}^T \mathbf{x}$$

onde $b = w_0$ e $\mathbf{w} = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix}^T$

- Matematicamente, é mais conveniente considerar $b = 0$ e incluir o atributo constante $x_0 = 1$ como parte do vetor $\mathbf{x}$:

$$\hat{y} = g(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

onde $\mathbf{w} = \begin{bmatrix} w_0 & w_1 & \cdots & w_n \end{bmatrix}^T$ e $\mathbf{x} = \begin{bmatrix} 1 & x_1 & \cdots & x_n \end{bmatrix}^T$

  - Usaremos essa notação daqui para frente, exceto quando mencionado o contrário

Função Custo

- Como $g(\cdot)$ é parametrizada por $\mathbf{w}$, denotamos a função custo do treinamento (erro médio no conjunto de treinamento) por

$$J(\mathbf{w}) = J(g) = \frac{1}{m}\sum_{i=1}^{m} L(g(\mathbf{x}^{(i)}), y^{(i)}) = \frac{1}{m}\sum_{i=1}^{m} L(\mathbf{w}^T\mathbf{x}^{(i)}, y^{(i)})$$

- Assumindo como função perda o erro quadrático $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$:

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})^2$$

- Como encontrar $\mathbf{w}$ que minimiza $J(\mathbf{w})$?

Função Custo

- Como $g(\cdot)$ é parametrizada por $\mathbf{w}$, denotamos a função custo do treinamento (erro médio no conjunto de treinamento) por

$$J(\mathbf{w}) = J(g) = \frac{1}{m}\sum_{i=1}^{m} L(g(\mathbf{x}^{(i)}), y^{(i)}) = \frac{1}{m}\sum_{i=1}^{m} L(\mathbf{w}^T\mathbf{x}^{(i)}, y^{(i)})$$

- Assumindo como função perda o erro quadrático $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$:

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})^2$$

- Como encontrar $\mathbf{w}$ que minimiza $J(\mathbf{w})$?
- **R:** Fazendo $\frac{\partial J(\mathbf{w})}{\partial w_j} = 0$

Otimização dos Parâmetros do Modelo

- Pode-se mostrar que

$$\nabla J(\mathbf{w}) = \begin{bmatrix} \frac{\partial J(\mathbf{w})}{\partial w_0} \\ \vdots \\ \frac{\partial J(\mathbf{w})}{\partial w_n} \end{bmatrix} = \frac{1}{m}\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

onde

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ \vdots \\ (\mathbf{x}^{(m)})^T \end{bmatrix} \quad \text{e} \quad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

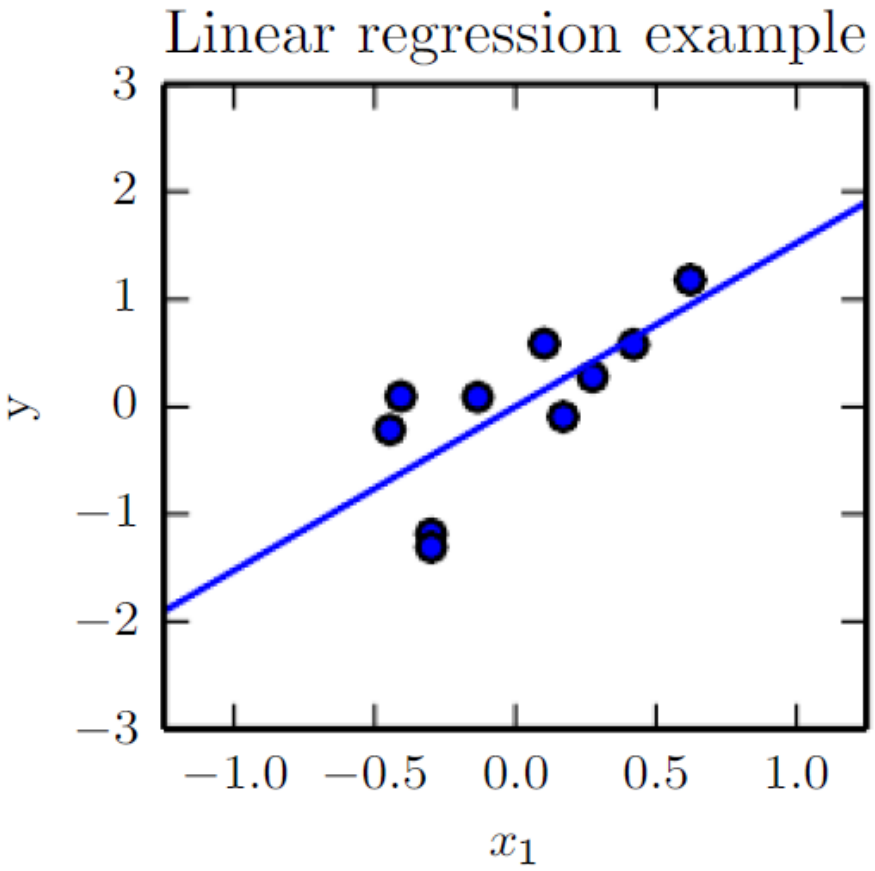
são a matriz de projeto (*design matrix*) e o vetor de rótulos de treinamento

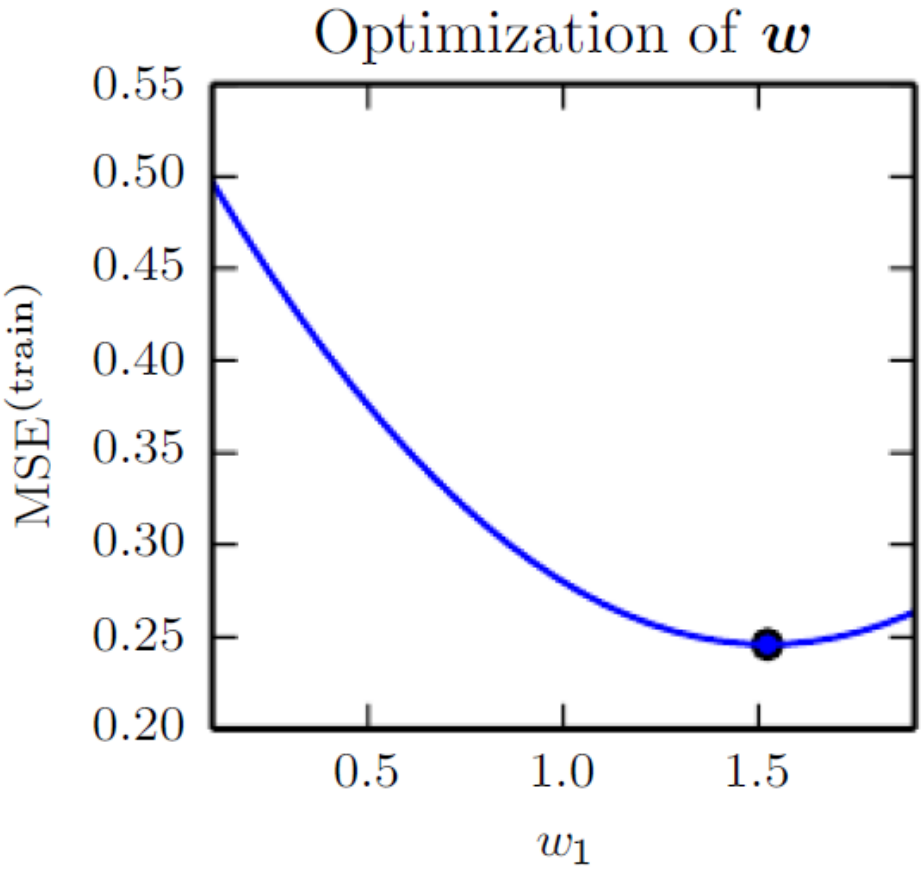
- Decorre que o valor ótimo de $\mathbf{w}$ é dado por

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

  - Conhecida como equação normal (*normal equation*)

Exemplo


Linear regression example
3
2
1
0
−1
−2
−3
y
−1.0
−0.5
0.0
0.5
1.0
x_1


Optimization of $\boldsymbol{w}$
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
$\mathrm{MSE}^{(\mathrm{train})}$
0.5
1.0
1.5
w_1

Interpretação Geométrica

- Predição no conjunto de treinamento: $\hat{y}^{(i)} = \mathbf{w}^T \mathbf{x}^{(i)} = {\mathbf{x}^{(i)}}^T \mathbf{w}$
- Seja $\hat{\mathbf{y}} = (\hat{y}^{(1)}, \ldots, \hat{y}^{(m)})^T = \mathbf{X}\mathbf{w}$ o vetor de predições
  - Note que $\hat{\mathbf{y}} \in \mathcal{S}$, onde $\mathcal{S}$ é o subespaço gerado pelas colunas de $\mathbf{X}$
- Erro de predição médio sobre o conjunto de treinamento:

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2 = \frac{1}{2m} \|\hat{\mathbf{y}} - \mathbf{y}\|^2$$

- Desejamos encontrar o vetor $\hat{\mathbf{y}} \in \mathcal{S}$ mais próximo de $\mathbf{y}$
- Solução é dada pela projeção ortogonal de $\mathbf{y}$ em $\mathcal{S}$:

$$\hat{\mathbf{y}} = \mathbf{P}\mathbf{y}$$

onde $\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ é a matriz de projeção

Regressão Linear com Funções de Base

- Suponha $n = 1$. Se desejarmos ajustar um modelo mais flexível do que uma reta $\hat{y} = w_0 + w_1 x$?
- Regressão polinomial:

$$\hat{y} = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

- A determinação de $\mathbf{w}$ continua a mesma: basta definir $x_i = x^i$, isto é, $\mathbf{x} = (1, x, x^2, x^3, \cdots, x^d)^T$ e $n = d$. Em particular:

$$\mathbf{X} = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^d \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x^{(m)} & (x^{(m)})^2 & \cdots & (x^{(m)})^d \end{bmatrix}$$

Regressão Linear com Funções de Base

- Em geral, podemos considerar como atributos $x_i = \varphi_i(x)$, i.e.,

$$\hat{y} = w_0 + w_1\varphi_1(x) + w_2\varphi_2(x) + \cdots + w_n\varphi_n(x)$$

  onde $\varphi_i(x)$ são funções de base quaisquer

- A determinação de $\mathbf{w}$ não muda pois o modelo continua sendo linear nos parâmetros:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- Do ponto de vista computacional, não importa se x_i é um atributo "natural" ou produzido "artificialmente" a partir de outros atributos

- Escolher bons atributos (ou transformações de atributos) é um problema de *feature engineering*

Limitações das Equações Normais

- O aumento de n introduz alguns desafios:
  - Para que $\mathbf{X}^T\mathbf{X}$ seja inversível, é necessário que $m \geq n$
  - Mesmo que $\mathbf{X}^T\mathbf{X}$ seja inversível, invertê-la pode ser computacionalmente custoso: a ordem de complexidade (em número de operações) é $O(n^3)$
- Essas limitações podem ser eliminadas usando um método de otimização iterativo: o método do gradiente (*gradient descent*)

Método do Gradiente

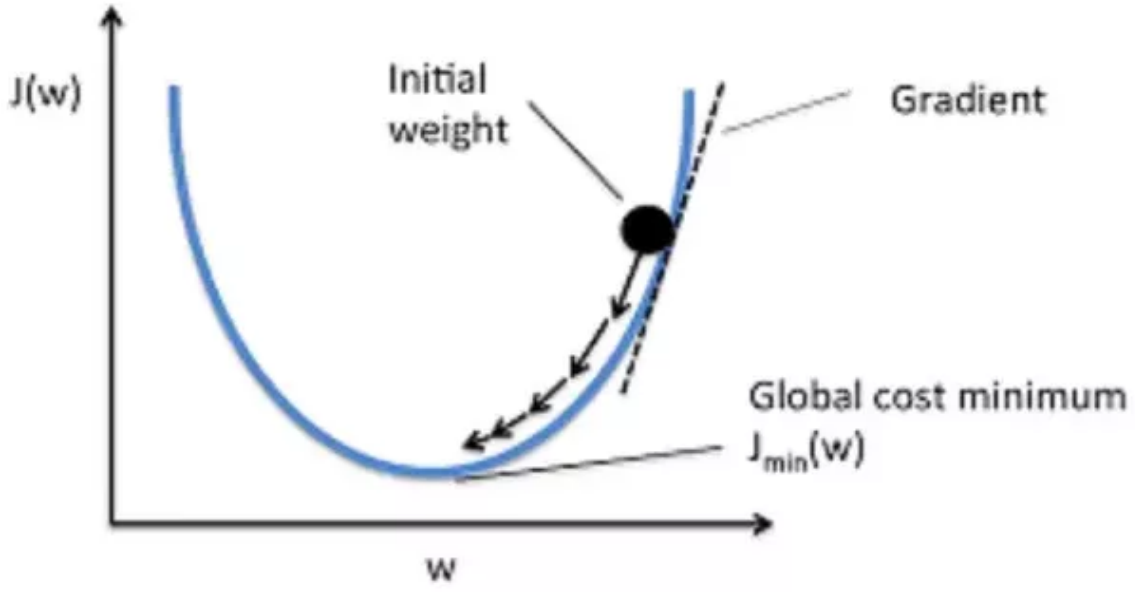
- Percorre o espaço de busca escolhendo sempre a direção de maior declive na função objetivo
- Método:
  - Inicie com pesos quaisquer $\mathbf{w}^{[0]}$
  - A cada iteração $t = 1, 2, \ldots$, atualize

$$\mathbf{w}^{[t+1]} = \mathbf{w}^{[t]} - \alpha \nabla J(\mathbf{w}^{[t]})$$

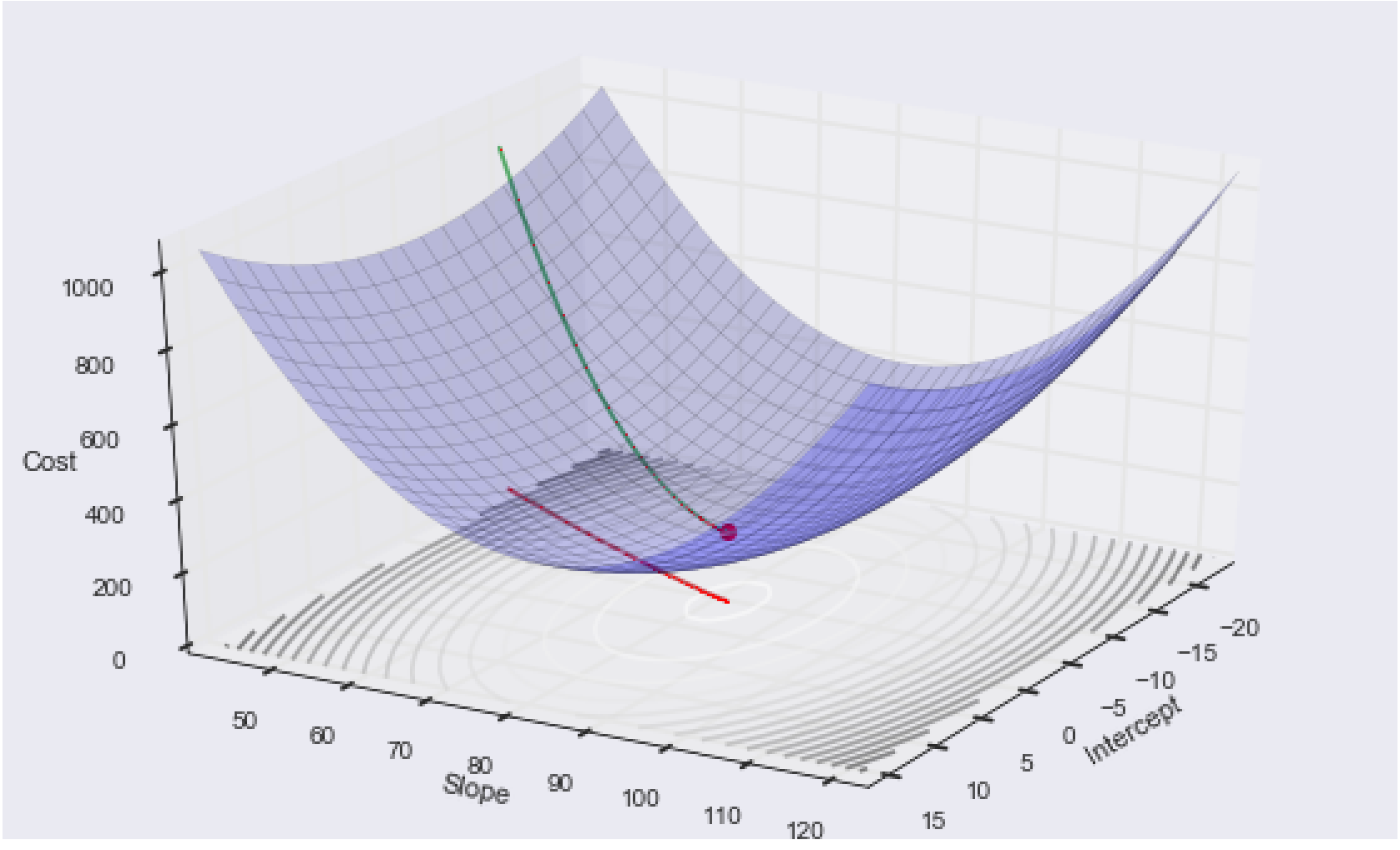
    onde α é a taxa de aprendizado
  - Se $\mathbf{w}^{[t]}$ praticamente não variar entre iterações (convergência), pare
- No caso em que $J(\mathbf{w})$ é o erro quadrático médio:

$$\mathbf{w}^{[t+1]} = \mathbf{w}^{[t]} - \alpha \frac{1}{m} \mathbf{X}^T (\mathbf{X}\mathbf{w}^{[t]} - \mathbf{y})$$

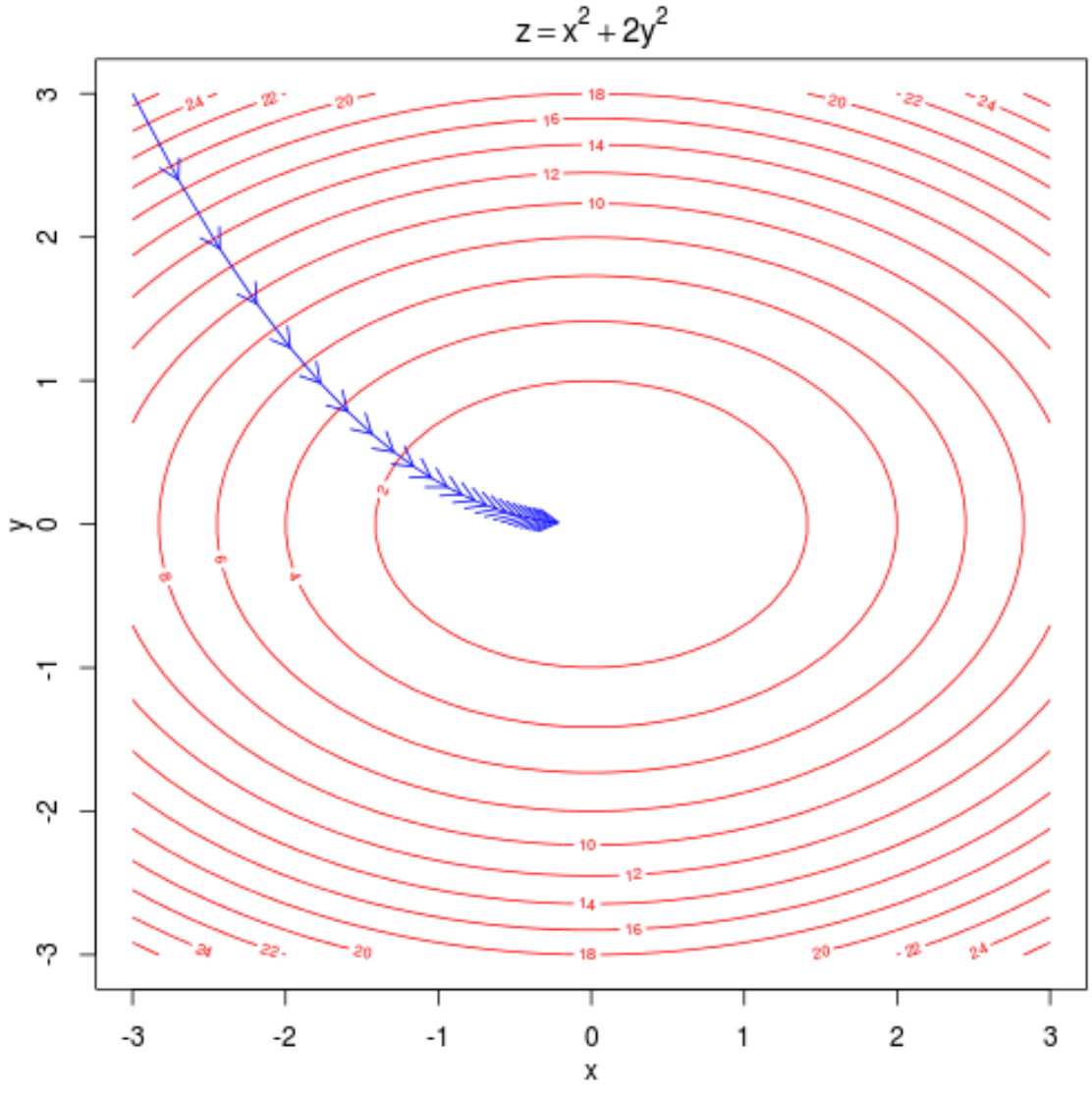
Exemplo



Exemplo


Cost
1000
800
600
400
200
0
Slope
50
60
70
80
90
100
110
120
Intercept
15
10
5
0
−5
−10
−15
−20

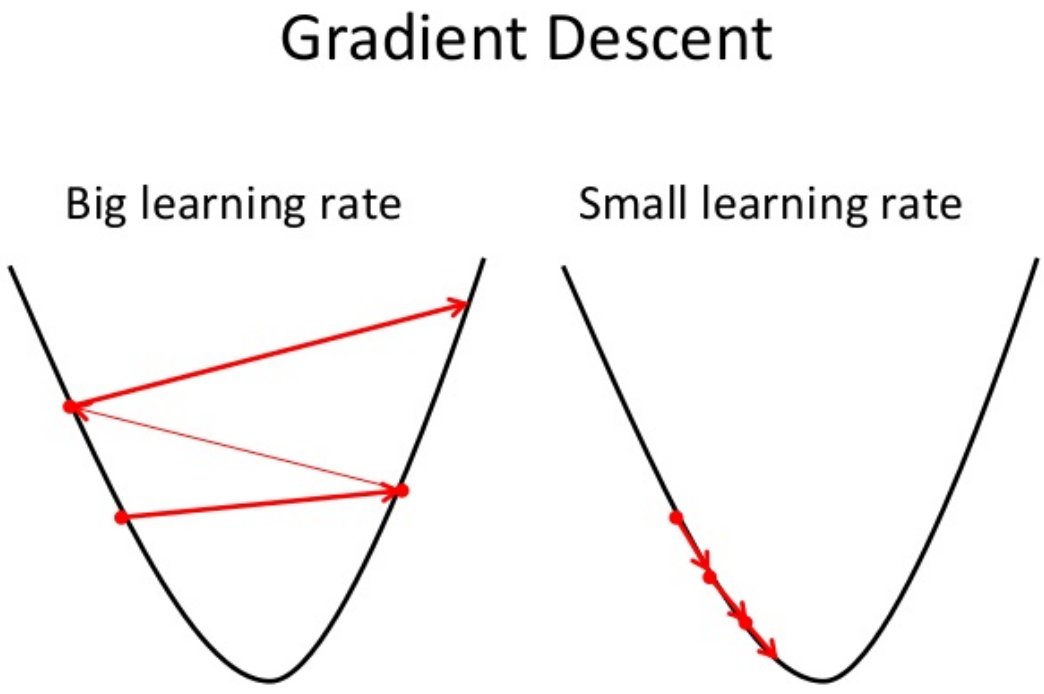
Exemplo



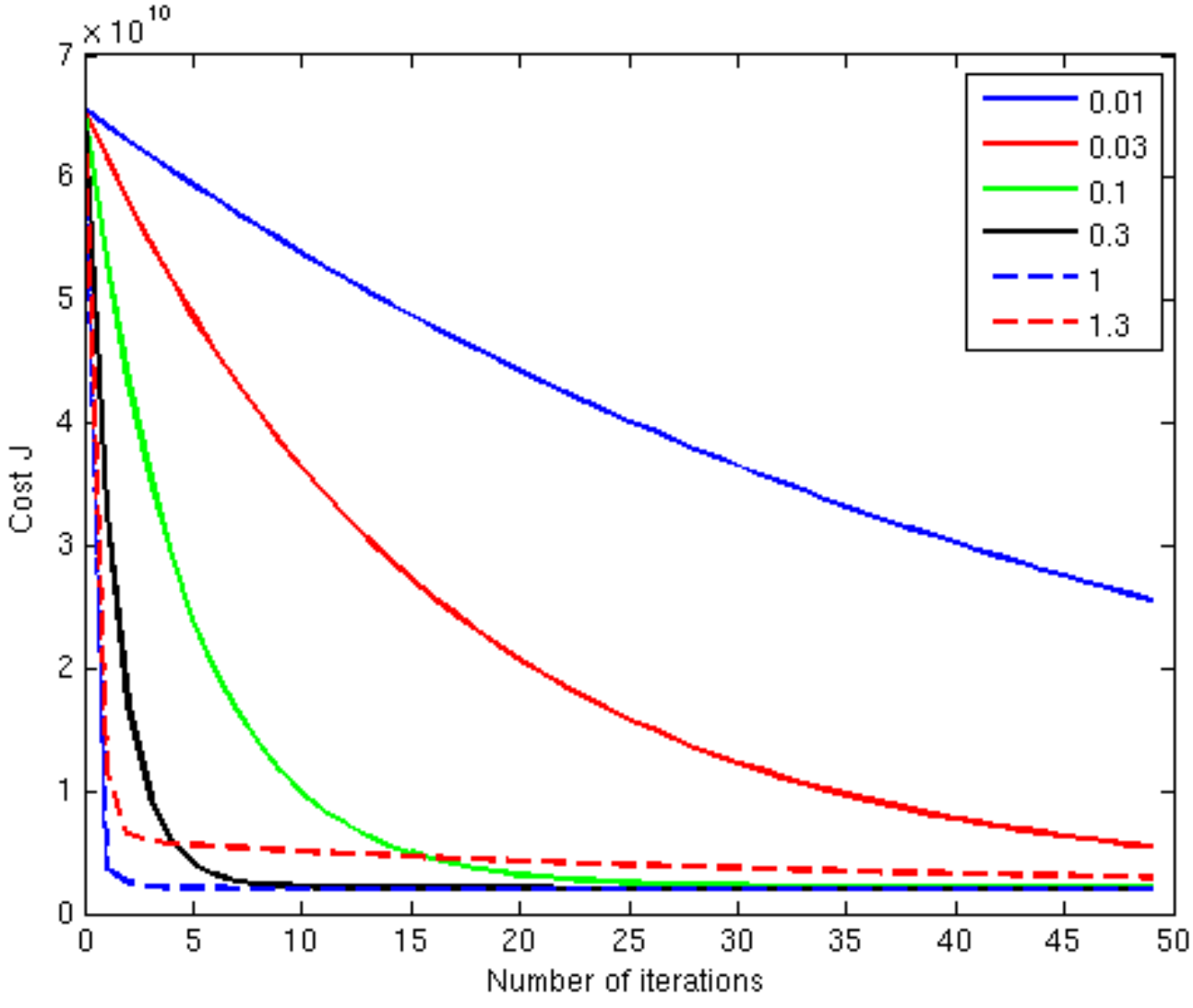
Escolha da Taxa de Aprendizado

- Uma das desvantagens do método do gradiente é ter de escolher a taxa de aprendizado
- Se α é muito pequeno, a convergência pode ser lenta
- Se α é muito grande, pode ocorrer overshoot. Nesse caso, o método pode não convergir ou até mesmo divergir
- Sempre é possível encontrar um valor de α (suficientemente pequeno) que garante convergência. No entanto, para acelerar a convergência na prática, também é possível usar um valor de α selecionado adaptativamente de acordo com a iteração, $\alpha^{[t]}$.

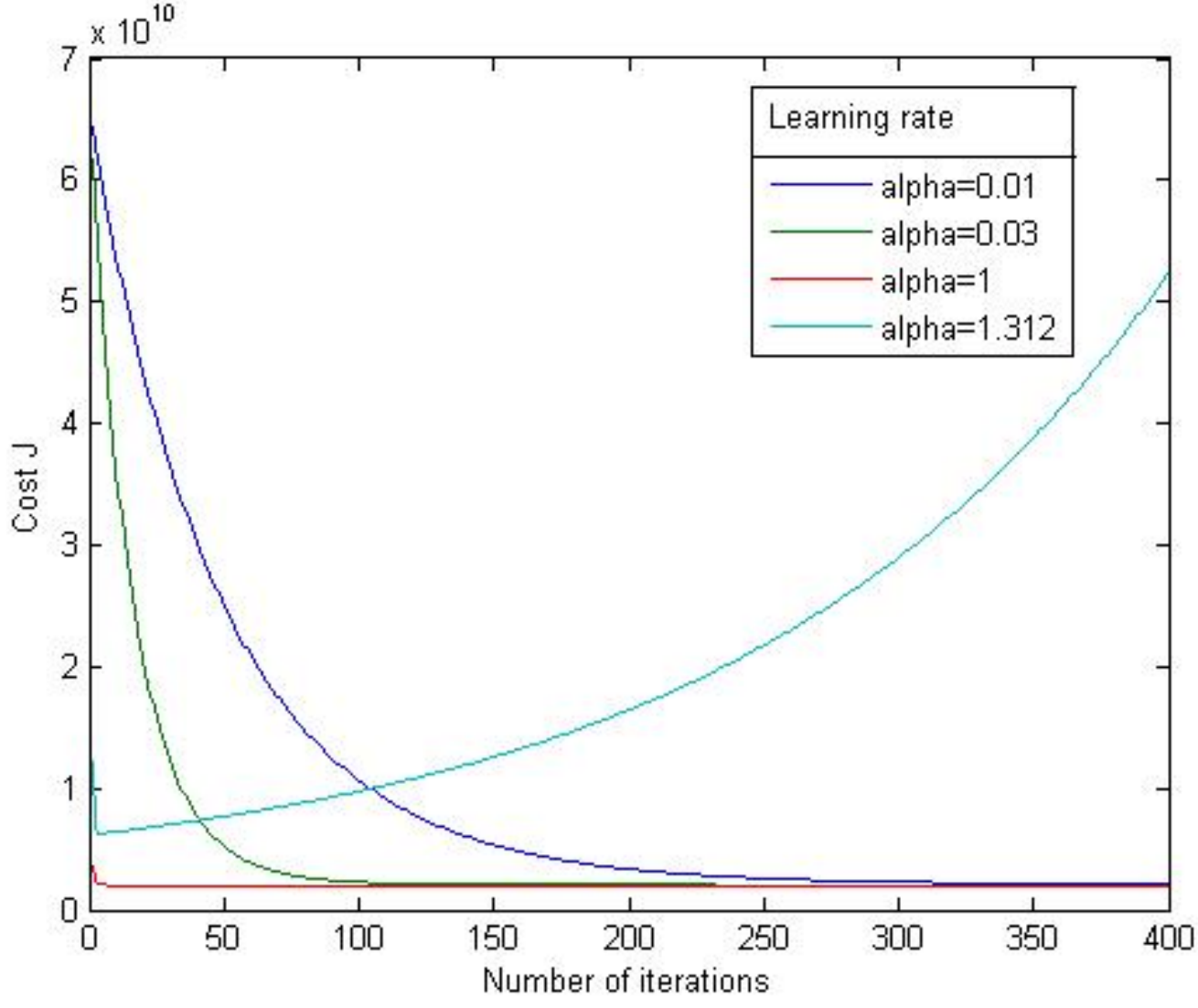
Exemplo



Exemplo: Custo em função da iteração



Exemplo: Custo em função da iteração



Exemplo: Custo em função da iteração