

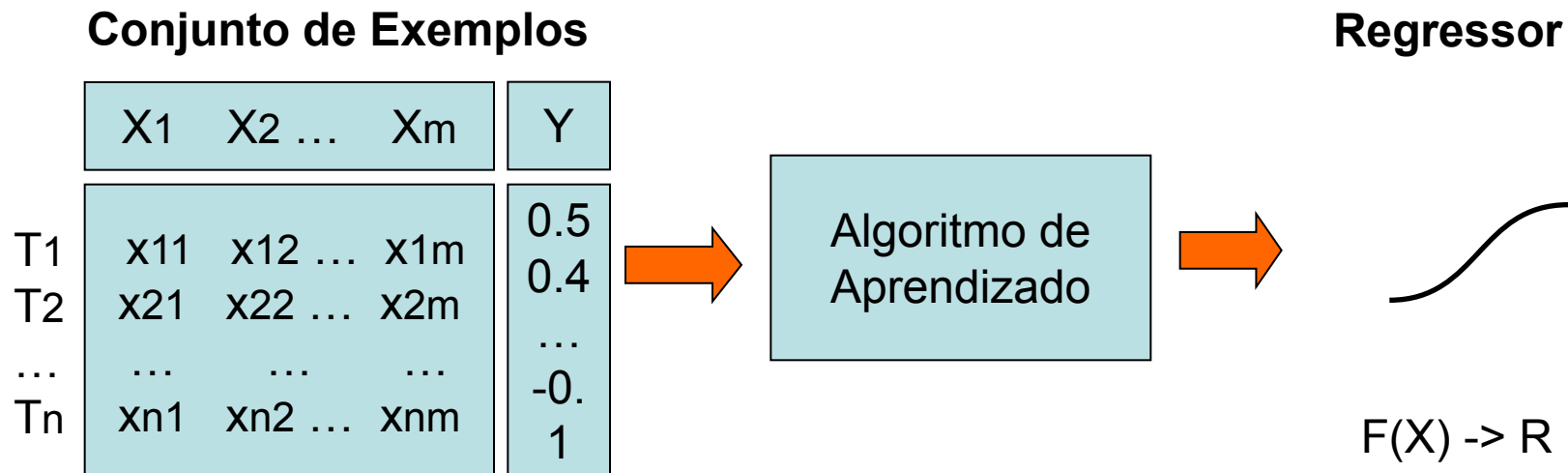
Algoritmos de Regressão

Ricardo Prudêncio

Regressão

- Associar exemplos a **valores numéricos**
 - E.g., previsão de índices da bolsa de valores, predição de custo de desenvolvimento de software,...
 - Similar à classificação, porém **atributo alvo é numérico**

Regressão




Regressão Linear

- Atributo alvo predito como uma **combinação linear** dos preditores

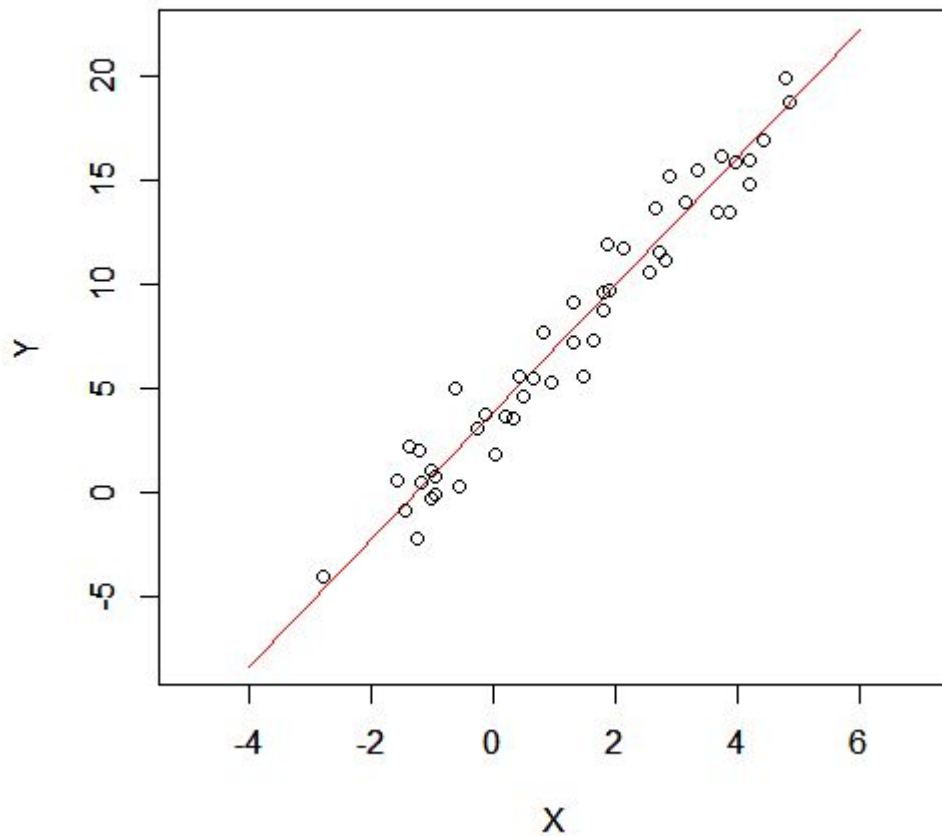
$$y = \beta + \sum_{i=1}^p \alpha_i x_i$$



Intercepto

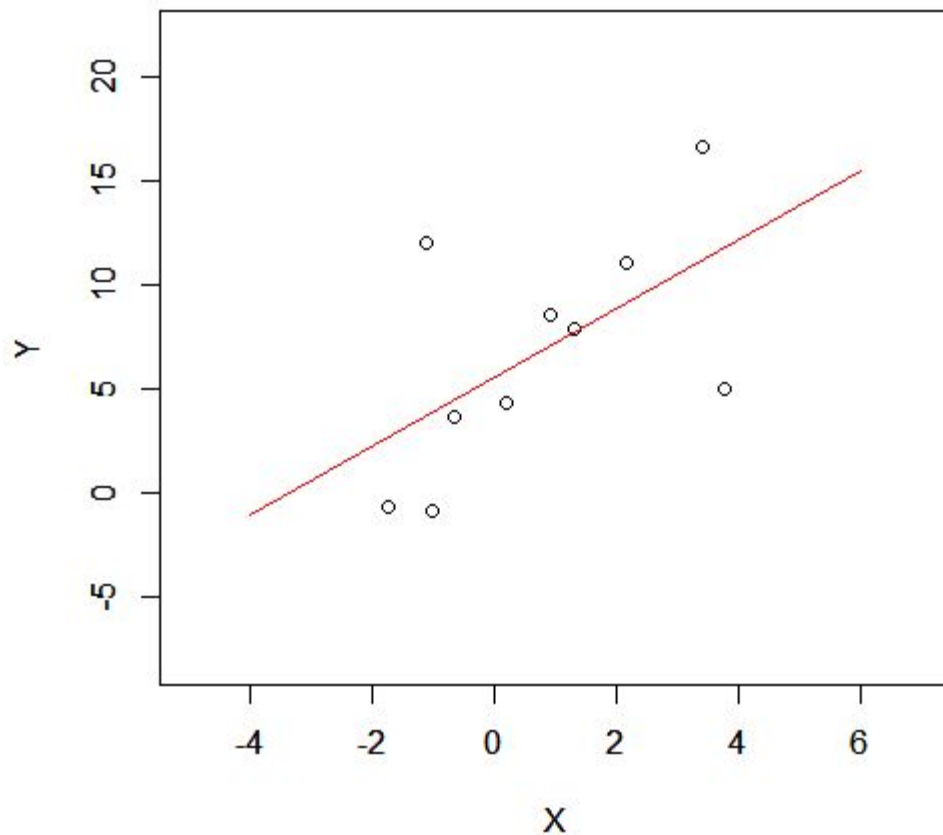


Coeficientes
das variáveis



$$\hat{\beta}, \hat{\alpha} = \arg \min \frac{1}{N} \sum_{i=1}^N (y_i - (\beta + \alpha x_i))^2$$

Método OLS – Ordinary Least Square – para estimativa dos parâmetros



Cuidado: ruído e outliers podem afetar muito o modelo

Uso de **métodos robustos**

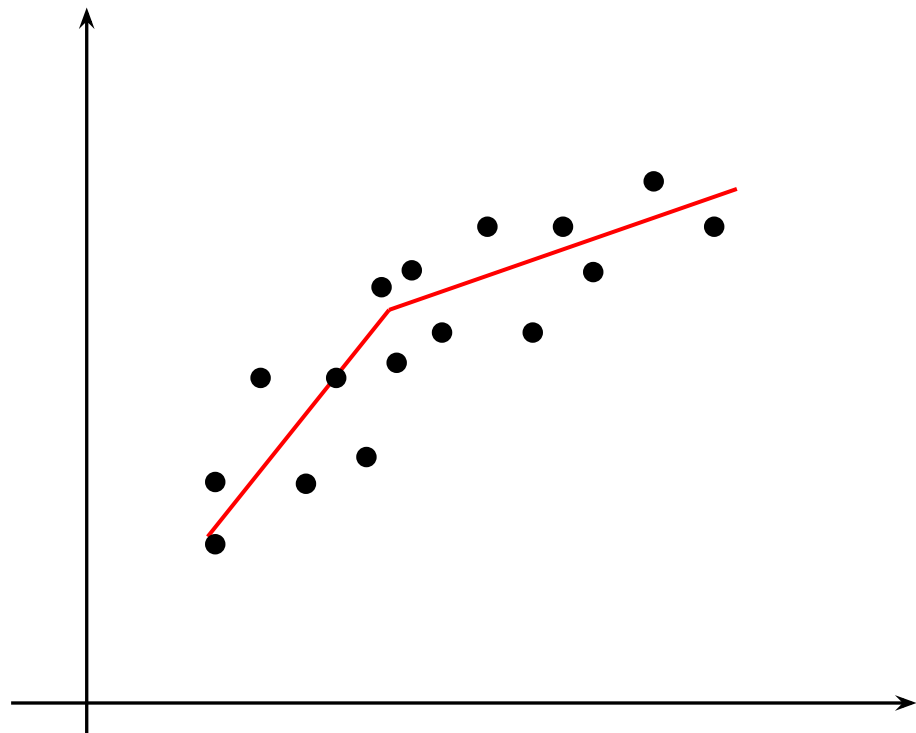
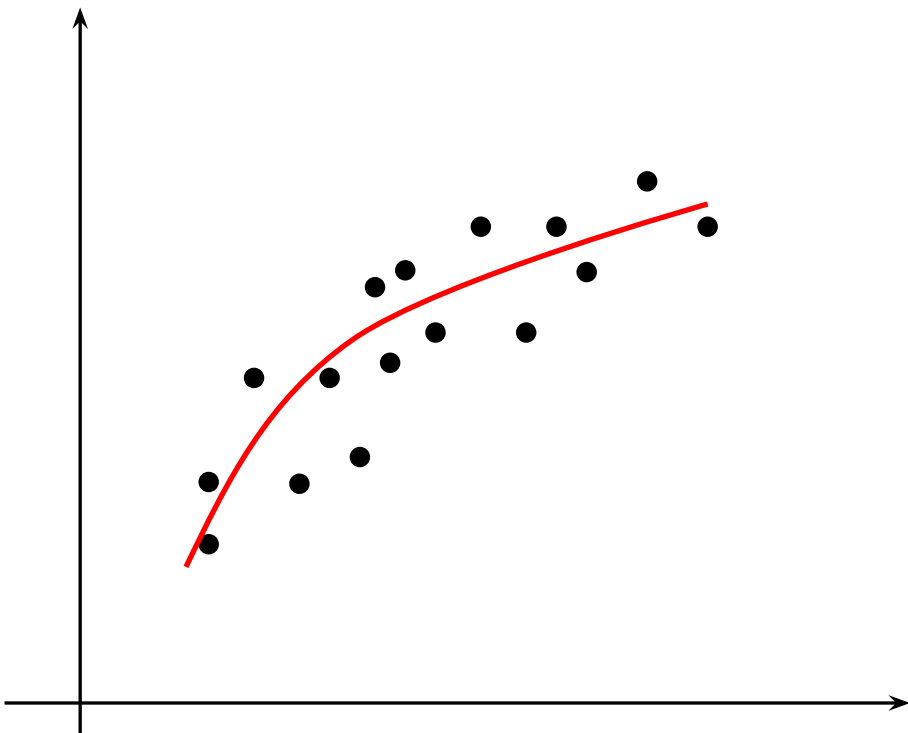
- Ridge Regression
- Lasso Regression

$$\hat{w} = \arg \min \frac{1}{N} \sum_{i=1}^N (y_i - wx_i)^2 + \frac{1-\lambda}{2} \|w\|_2^2 + \lambda \|w\|_1$$

Árvores de Regressão

- Similar às árvores de decisão usadas para classificação
- Critério de separação para atributo-alvo numérico
- Predição nas folhas é feita através da média ou de um modelo local

Árvores de Regressão



Árvores de Regressão

- Construção

- Árvore de decisão construída de forma *recursiva* da raiz para as folhas (*top-down*)
 - A cada nó, é escolhido um *teste* que separe melhor os exemplos de classes diferentes
 - Maximização de *critério de separação*
 - Nós terminais são criados ao atingir um *critério de parada*
 - Ex.: a variância do atributo alvo no nó é baixa

Árvores de Regressão

- Construção

- AD(Exemplos:D; Atributos:A; Alvo:C)
 - Crie nó_raiz
 - SE *Critério_de_Parada*
ENTÃO Crie nó terminal associado a um **modelo local**
 - SENÃO Encontre *atributo aj* cujo *teste de decisão* diminua a *variação do atributo alvo* para os exemplos que atingem o nó
 - PARA CADA *valor v* do teste adicione
nova sub-árvore
 - Sub_arvore = AD($D_{[a_j = v]}$, $A - \{a_j\}$, C)

Árvores de Decisão

– Critérios de Separação

- Erro Quadrado Médio
 - Erro obtido pelo *regressor simples (média)*

$$Err(D) = \frac{1}{|D|} \sum_{i \in D} (y_i - \bar{y}_D)^2$$

$$\bar{y}_D = \frac{1}{|D|} \sum_{i \in D} y_i$$

Árvores de Decisão

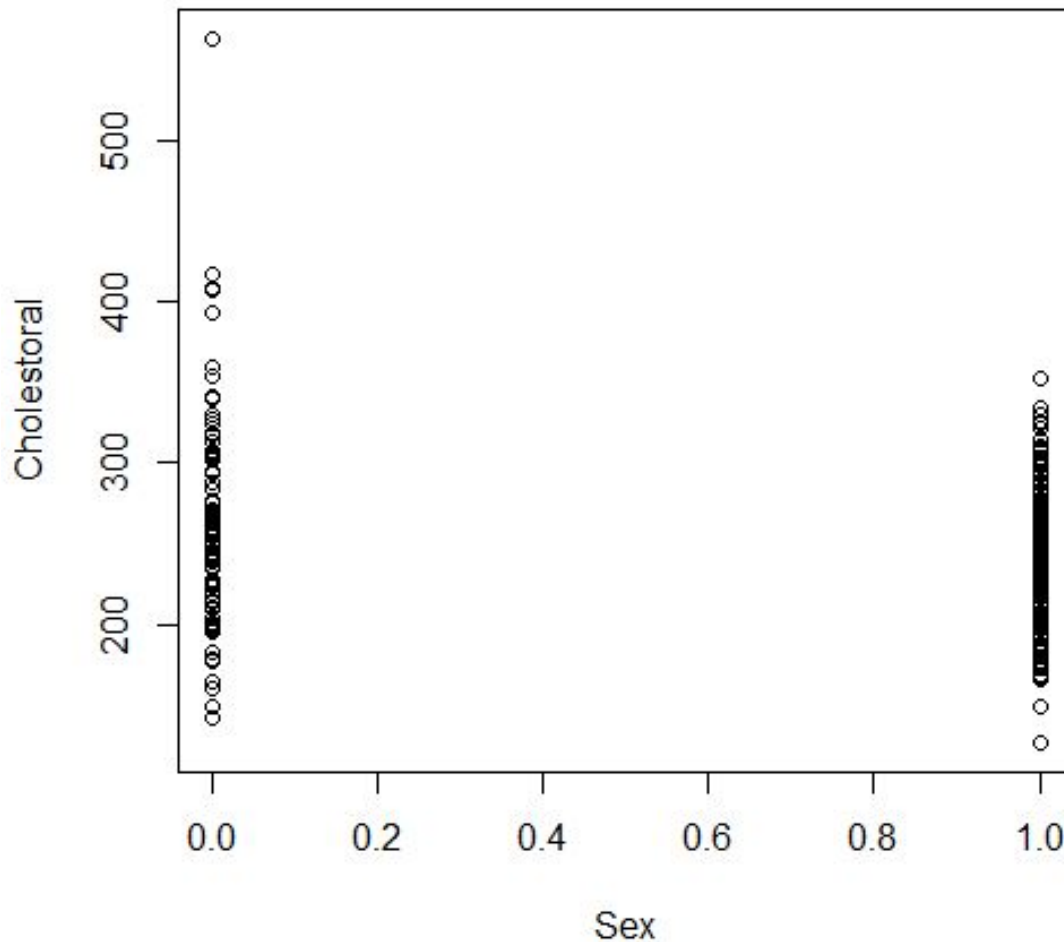
– Critérios de Separação

- Diminuição do Erro Quadrado Médio

$$\Delta Err(D, a) = Err(D) - Err(D, a)$$

$$Err(D, a) = \sum_{v_i} \frac{|D_{[a=v_i]}|}{|D|} * Err(D_{[a=v_i]})$$

↑
EQM no nó filho



Cholestoral

270 exemples
Média = 249.65
Err = 2661.57

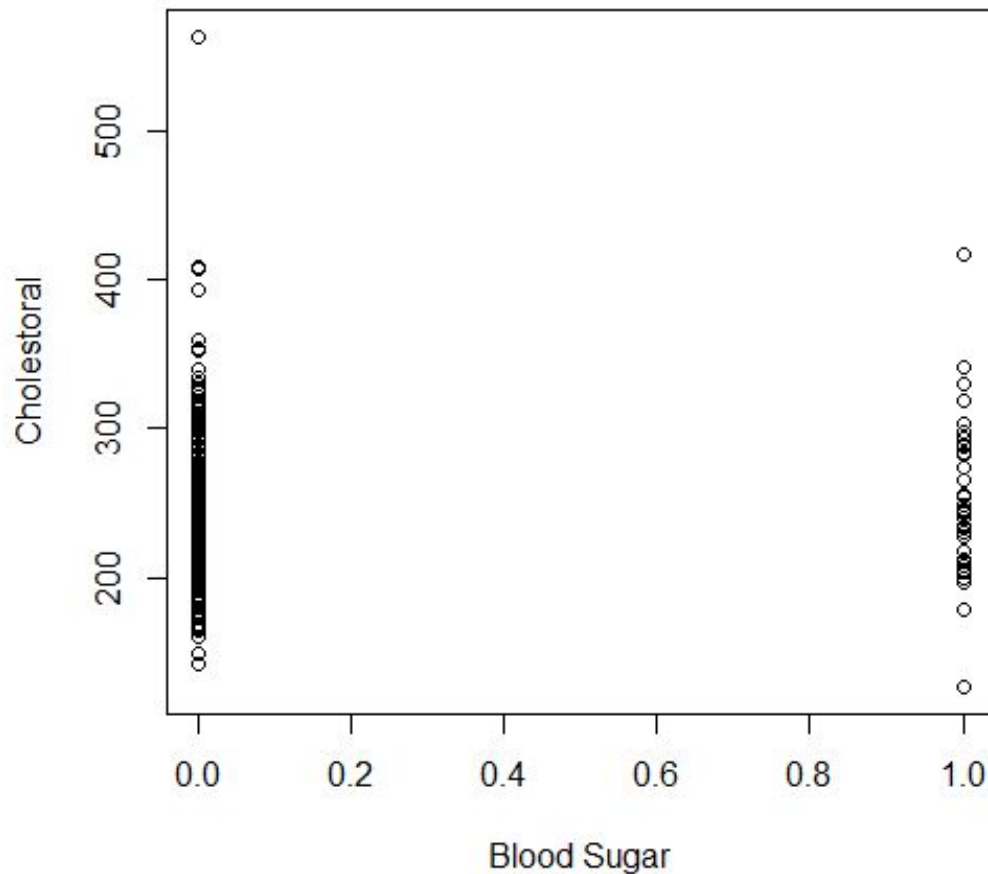
Sex = 0

- 87 exemples
- Média Chol. = 264.74
- Err = 4356.23

Sex = 1

- 183 exemples
- Média Chol. = 242.48
- Err = 1696.23

$$\text{Err}(D) - \text{Err}(D, \text{sex}) = 2661.57 - (87/270 \cdot 4356.23 + 183/270 \cdot 1696.23) \\ = \mathbf{108.22}$$



Cholestoral

270 exemples

Média = 249.65

Err = 2661.57

Sugar = 0

- 230 exemples

- Média Chol. = 249.11

- Err = 2672.99

Sugar = 1

- 40 exemples

- Média Chol. = 252.77

- Err = 2584.47

$$\begin{aligned} \text{Err}(D) - \text{Err}(D, \text{sex}) &= 2661.57 - (230/270 * 2672.99 + 40/270 * 2584.47) \\ &= \mathbf{1.68} \end{aligned}$$

Árvores de Decisão

– Nós Terminais

- Modelo local para predição
 - Regressor simples (média)
 - Predição = média do atributo alvo dos exemplos de treinamento que atingem o nó terminal
 - Regressão linear simples
 - Aprendido com os exemplos de treinamento que atingem o nó terminal

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose M5P -M 4.0

Test options

☐ Use training set☐ Supplied test set

Set...

☒ Cross-validation

Folds

10

☐ Percentage split

%

66

More options...

(Num) chol

Start

Stop

Result list (right-click for options)

17:15:16 - trees.M5P

Classifier output

```
thal
num
chol
Test mode:10-fold cross-validation
=== Classifier model (full training set) ===

M5 pruned model tree:
(using smoothed linear models)

sex=0 <= 0.5 : LM1 (206/80.996%)
sex=0 > 0.5 : LM2 (97/110.912%)

LM num: 1
chol =
  0.0741 * age
+ 1.646 * sex=0
+ 1.0892 * restecg=2,1
+ 0.0158 * thalach
+ 7.4293 * ca
+ 0.9076 * thal=7
+ 226.8278

LM num: 2
chol =
  1.9344 * age
+ 3.2479 * sex=0
+ 27.5845 * restecg=2,1
+ 0.7142 * thalach
+ 39.5547 * thal=7
+ 22.9485

Number of Rules : 2

Time taken to build model: 1.22 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient      0.1874
Mean absolute error         39.8923
Root mean squared error     52.2994
Relative absolute error     101.1056 %
Root relative squared error 100.8867 %
Total Number of Instances   303
```

Modelo de regressão linear
nas folhas

Status

OK

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose M5P -R -M 4.0

Test options

☐ Use training set☐ Supplied test set

Set...

☒ Cross-validation Folds 10☐ Percentage split % 66

More options...

(Num) chol

Start

Stop

Result list (right-click for options)

17:15:16 - trees.M5P

17:49:17 - trees.M5P

Classifier output

```
num
chol
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

M5 pruned regression tree:
(using smoothed linear models)

sex=0 <= 0.5 : LM1 (206/82.308%)
sex=0 > 0.5 :
| age <= 54.5 : LM2 (40/85.04%)
| age > 54.5 :
| | thalach <= 149 : LM3 (25/91.53%)
| | thalach > 149 : LM4 (32/143.164%)

LM num: 1
chol =
    + 240.0832

LM num: 2
chol =
    + 245.2155

LM num: 3
chol =
    + 255.1392

LM num: 4
chol =
    + 284.1791

Number of Rules : 4

Time taken to build model: 0.3 seconds

=== Cross-validation ===
=== Summary ===

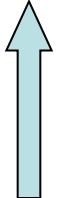
Correlation coefficient      -0.0331
Mean absolute error         40.4806
Root mean squared error     52.9426
Relative absolute error     102.5965 %
Root relative squared error 102.1273 %
Total Number of Instances   303
```

Predição pela média

Status
OK

Algoritmo k-NN para Regressão

- Algoritmo pode ser usado para estimar valores de funções contínuas

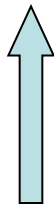
$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k f(x_i)}{k}$$


- Predição é a *média simples* dos valores alvo armazenados nas instâncias recuperadas

Algoritmo k-NN para Regressão

- Regressão com Ponderação pela Distância

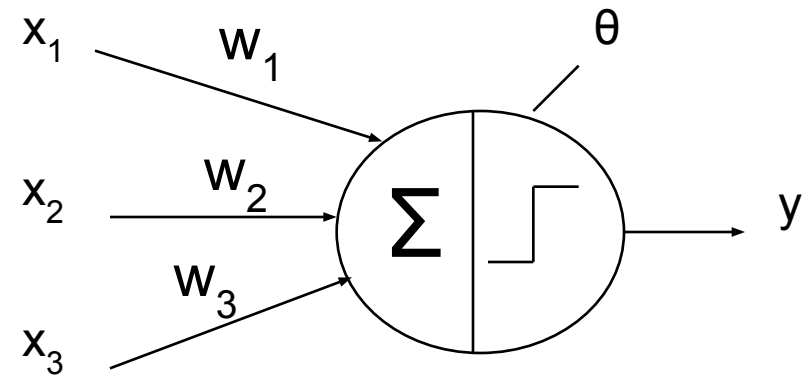
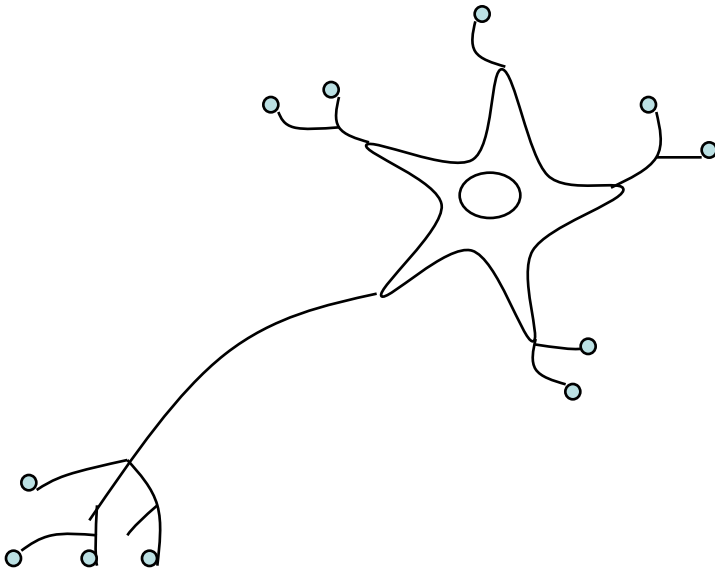
$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i}$$



- Predição é a *média ponderada* dos valores alvo armazenados nas instâncias recuperadas

Modelos de Redes Neurais

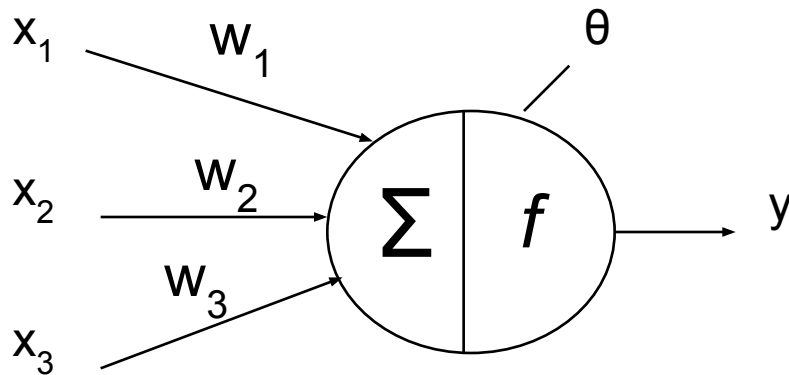
- Modelo matemático de um neurônio por Warren McCulloch e Walter Pitts (1943)



$$y = 1 \quad \text{se } \sum_i w_i * x_i > \theta$$
$$y = 0 \quad \text{caso contrário}$$

Modelos de Redes Neurais

- Modelo de neurônio contínuo

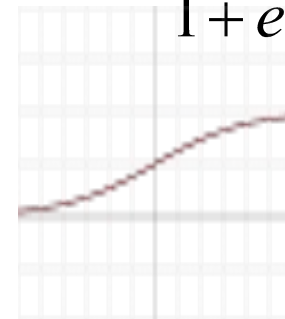


$$y = f(w * x - \theta)$$

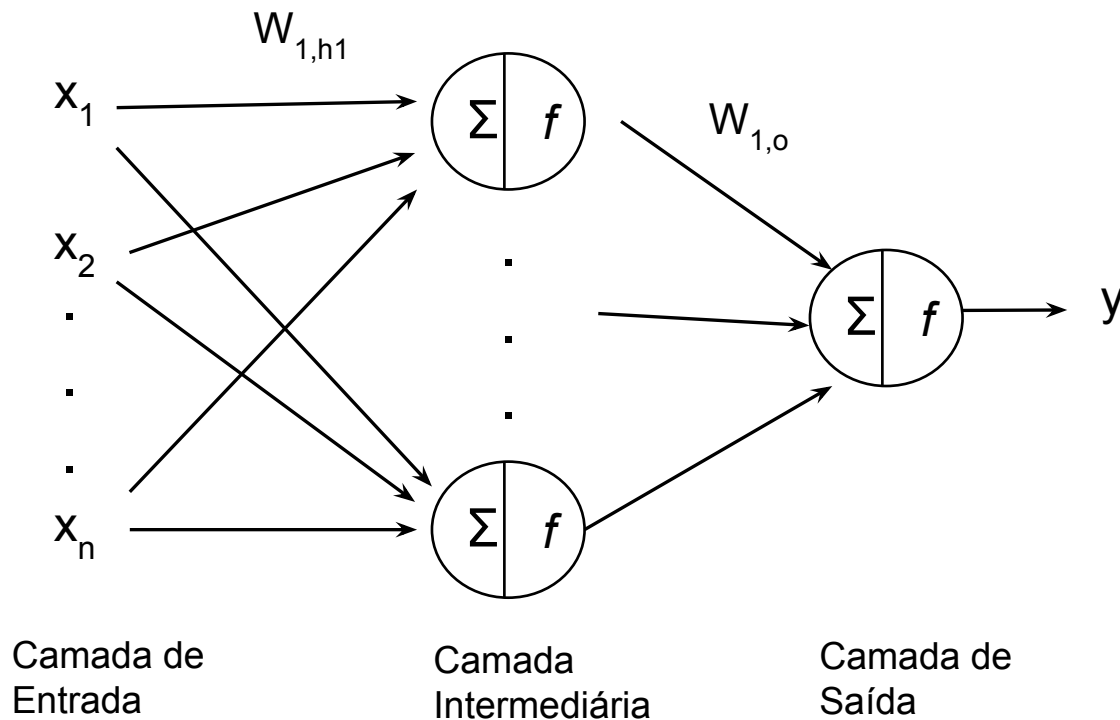
f : função de ativação

Exemplo:

$$f(x) = \frac{1}{1 + e^{-x}}$$




MultiLayer Perceptron (MLP)



MultiLayer Perceptron (MLP)

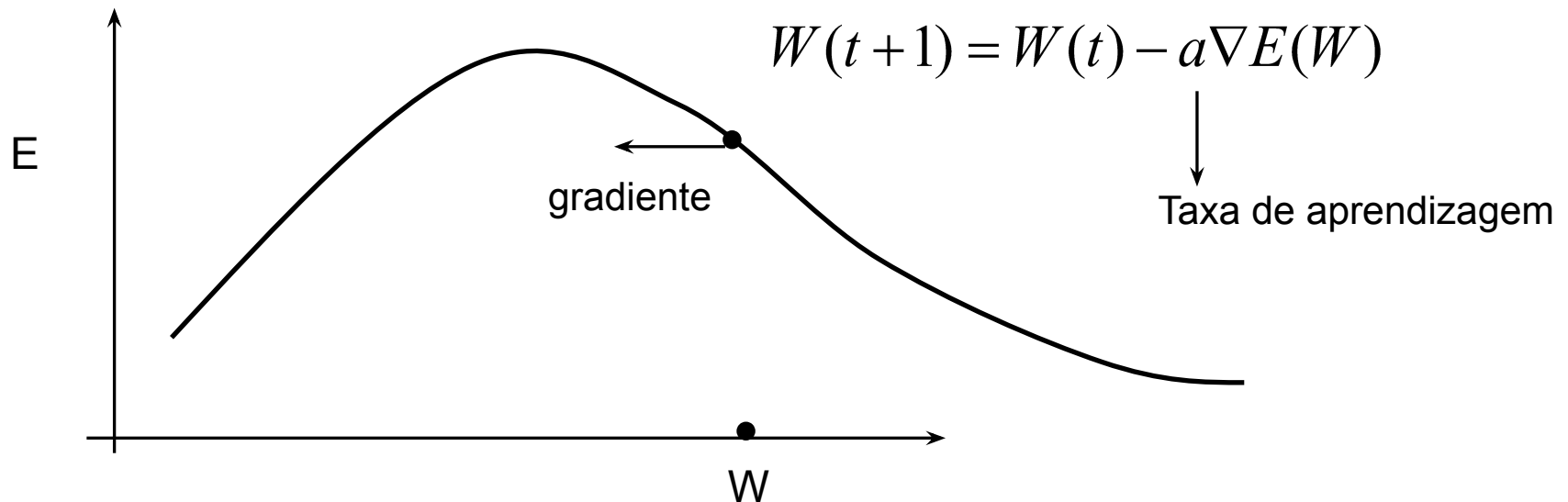
- Algoritmo Backpropagation
 - Aprendizado dos pesos das redes MLP
 - Minimização do erro quadrado da rede

$$Err(D) = \frac{1}{|D|} \sum_{i \in D} (y_i - f(x_i))^2$$


Saída da rede

Algoritmo Backpropagation

- Método do **gradiente descendente**
 - **Ajuste iterativo** dos pesos na direção em que a função de erro tem sua maior variação



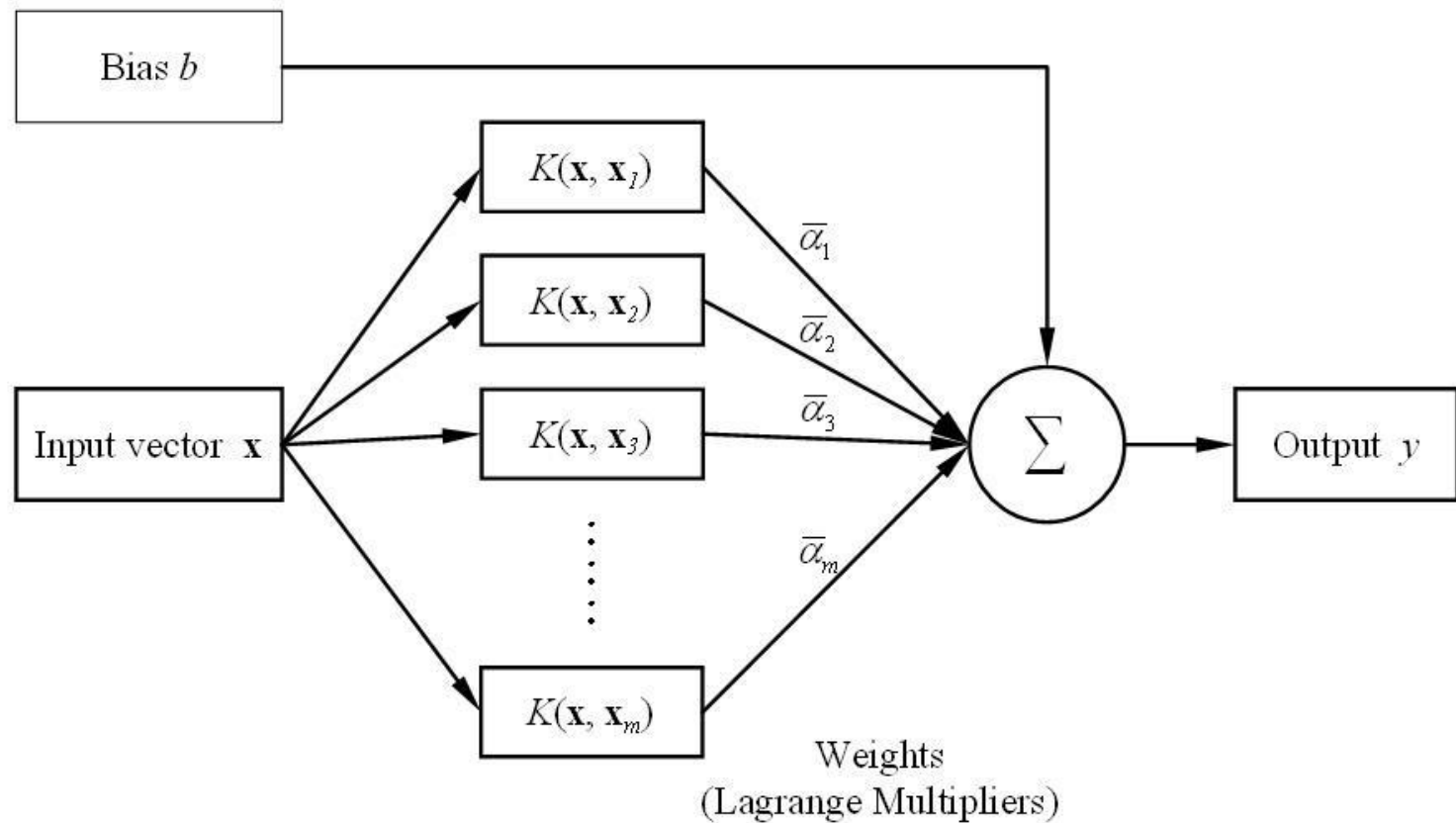
MultiLayer Perceptron (MLP)

- Dificuldades
 - Definição da arquitetura da rede
 - Número de neurônios intermediários
 - Funções de ativação
 - Algoritmo de backpropagation
 - Definição da taxa de aprendizagem
 - Mínimos locais da função de erro

Support Vector Machines

- Também realizam mapeamento do espaço de entradas
- Mapeamento realizado através de **funções de kernel**
- Aprendizado de SVMs usa **métodos de otimização** mais eficazes

Support Vector Machines



Hidden Nodes
(Support Vectors)

$$y = f(\mathbf{x}) = \sum_{k=1}^m \bar{\alpha}_k \cdot K(\mathbf{x}, \mathbf{x}_k) + b$$

SVM (Kernels)

- Função de Kernel
 - Exemplos

$$K(x_i, x_j) = (x_i x_j + 1)^p \longrightarrow \text{Kernel polinomial} \\ \text{(parâmetro } p)$$

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \longrightarrow \text{Kernel RBF} \\ \text{(parâmetro gamma)}$$

Support Vector Machines

- Vantagens
 - Definição da arquitetura é feita pelo próprio processo de aprendizado
 - Bons resultados empíricos
- Desvantagens
 - Definição adequada da função de kernel
 - Custo computacional

Material de Estudo

- T. Mitchell, *Machine Learning*, 1997.
- I. Witten, E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*. 2000.
- Ross J. Quinlan: Learning with Continuous Classes. In: 5th Australian Joint Conference on Artificial Intelligence, Singapore, 343-348, 1992.
- A. Smola. A Tutorial on Support Vector Regression. 2004.