Inteligência Artificial

Modelos Lineares

Prof. Fabio Augusto Faria

1º semestre 2021

UNIFESP
Universidade Federal de São Paulo
1933

Tópicos

- Regressão e Classificação
- Regressão Linear
- Classificador Linear com Limiar Rígido
- Classificador Linear com Regressão Logística

Aprendizagem Indutiva

- Dada uma coleção de exemplos de ***f***, retornar uma função ***h*** que se aproxime de ***f*.**
- **x** são exemplos de entrada
- ***h*** é a função de mapeamento buscada
- **f** é a função alvo/objetivo

Espera-se que:

f(x) = h(x)

para diferentes exemplos amostrados

“Melhor Ajuste”

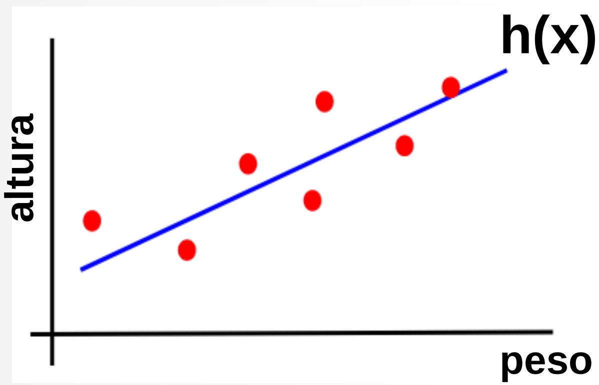
Aprendizagem Supervisionada

$f(x) - h(x)$, $f(x) = y$ e $h(x) = y'$ → ACERTO é: $y = y'$

Aprendizagem Supervisionada

f(x) – h(x), f(x) = y e h(x) = y’ → ACERTO é: y = y’

- Regressão: y é valor contínuo

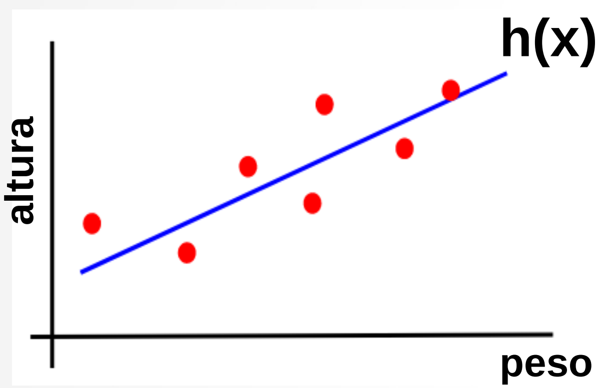


peso = 80kg, **qual valor de altura**?

Aprendizagem Supervisionada

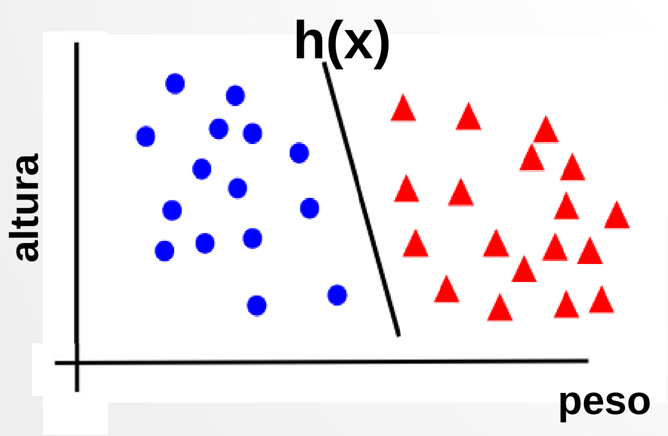
f(x) – h(x), f(x) = y e h(x) = y' → ACERTO é: y = y'

- Regressão: y é valor contínuo



peso = 80kg, **qual valor de altura**?

- Classificação: y é valor discreto/categórico

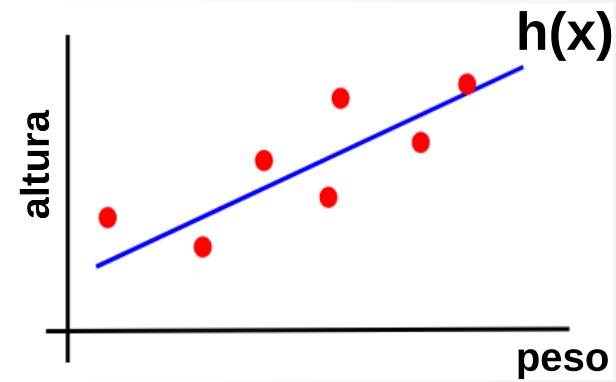


peso = 80kg e altura = 180, **IMC = BOM ou RUIM**?

Regressão Linear

- **Regressão Linear**: uso de linha/reta para ajustar aos dados de treinamento
- *Fórmula da reta é:*

$$h_{\mathbf{w}}(x) = w_1 x + w_0 \ .$$



- **Função de PerdaEmpírica L2 é:**

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 \ .$$

Objetivo é encontrar W0 e W1 que minimiza a L2.

Regressão Linear

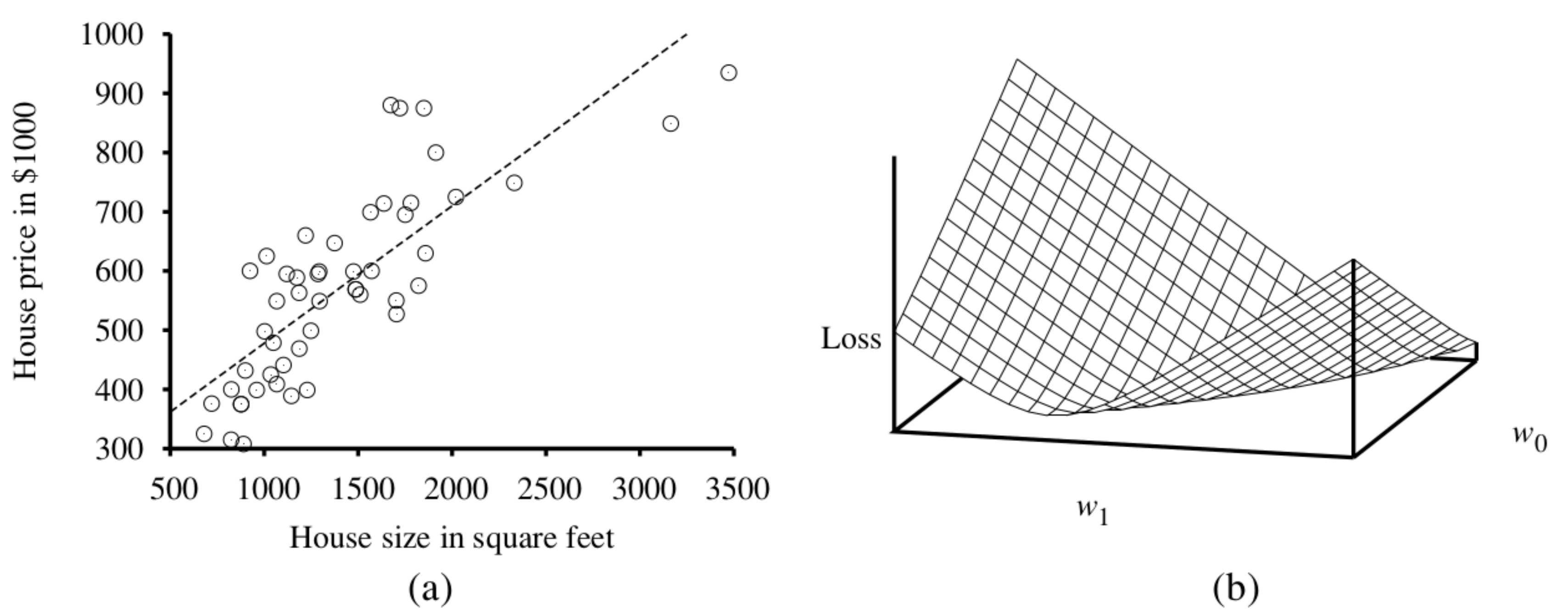


Figure 18.13 (a) Data points of price versus floor space of houses for sale in Berkeley, CA, in July 2009, along with the linear function hypothesis that minimizes squared error loss: $y = 0.232x + 246$. (b) Plot of the loss function $\sum_j (w_1 x_j + w_0 - y_j)^2$ for various values of w_0, w_1. Note that the loss function is convex, with a single global minimum.

Regressão Linear

- **Função de PerdaEmpírica L2 é:**

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 .$$

- **Minizar derivadas parciais:**

$$\frac{\partial}{\partial w_0} \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 = 0 \text{ and } \frac{\partial}{\partial w_1} \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 = 0 . \qquad (18.2)$$

- **Encontrar W1 e W0:**

$$w_1 = \frac{N(\sum x_j y_j) - (\sum x_j)(\sum y_j)}{N(\sum x_j^2) - (\sum x_j)^2}; \quad w_0 = (\sum y_j - w_1(\sum x_j))/N . \qquad (18.3)$$

Regressão Linear

- **Descida pelo Gradiente** **para atualização dos pesos W={w0,w1}:**

$$\mathbf{w} \leftarrow \text{any point in the parameter space}$$

loop until convergence **do**

  for each w_i **in w do**

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(\mathbf{w})$$

$$w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j))\,; \quad w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) \times x_j\,.$$

These updates make intuitive sense: if $h_{\mathbf{w}}(x) > y$, i.e., the output of the hypothesis is too large, reduce w_0 a bit, and reduce w_1 if x was a positive input but increase w_1 if x was a negative input.

Regressão Linear Multivariada

- **Regressão Linear**: uso de linha/reta para ajustar aos dados de treinamento
- *Fórmula é:*

$$h_{sw}(\mathbf{x}_j) = w_0 + w_1 x_{j,1} + \cdots + w_n x_{j,n} = w_0 + \sum_i w_i x_{j,i} \,.$$

$$h_{sw}(\mathbf{x}_j) = \mathbf{w} \cdot \mathbf{x}_j = \mathbf{w}^\top \mathbf{x}_j = \sum_i w_i x_{j,i} \,.$$

Objetivo é encontrar W={w0,w2,w3,…,wi}.

- **Função de PerdaEmpírica L2 é:**

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_j L_2(y_j, \mathbf{w} \cdot \mathbf{x}_j) \,.$$

Regressão Linear Multivariada

- **Descida pelo Gradiente** para atualização dos pesos W={w0,w1,w2,…,wi}:

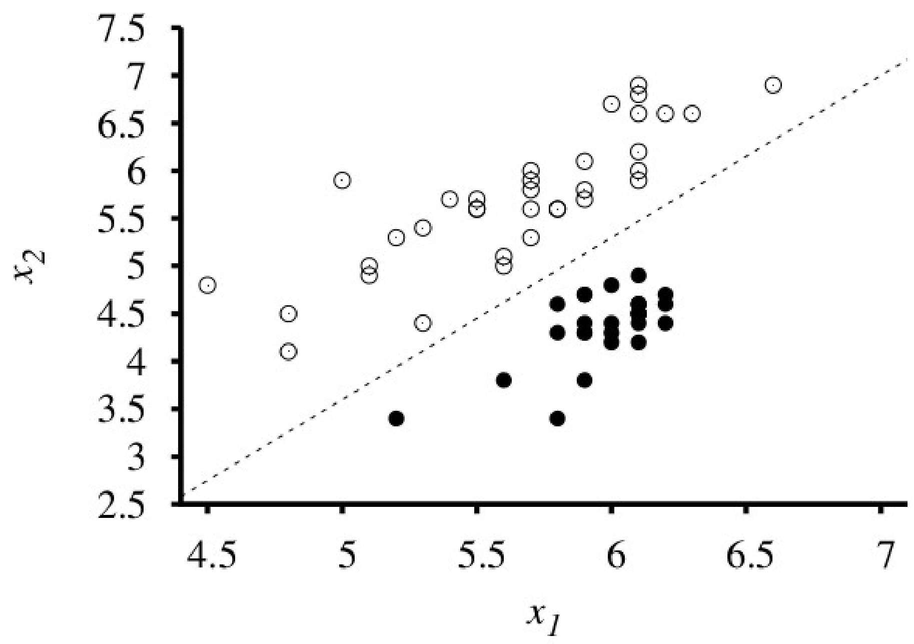
$$w_i \leftarrow w_i + \alpha \sum_j x_{j,i}(y_j - h_{\mathbf{w}}(\mathbf{x}_j)) \; .$$

- **y é um vetor com as saídas para as amostras de treinamento e X é uma matriz de dados (dados de entrada com n dimensões:**

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

Classificador Linear com Limiar Rígido

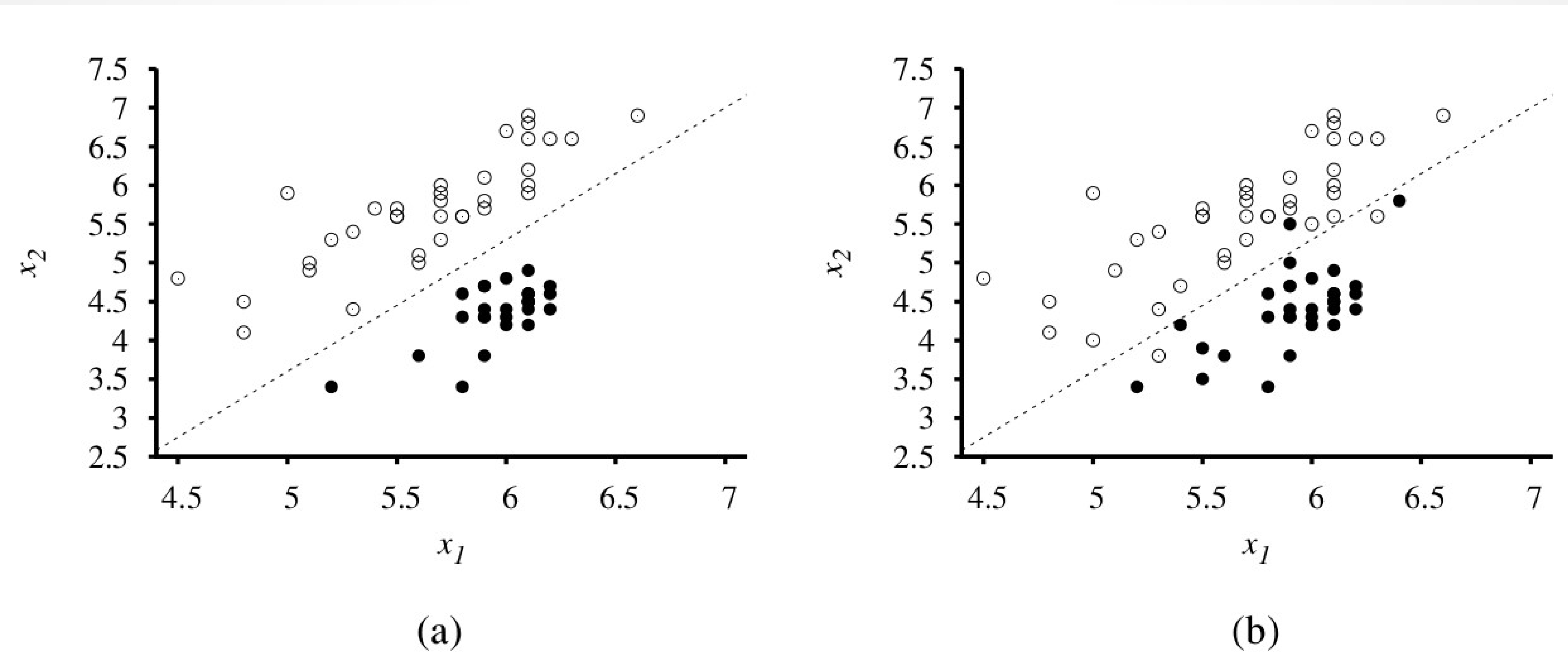
- **Classificador Linear**: uso de linha/reta para separar os dados de treinamento



(a)

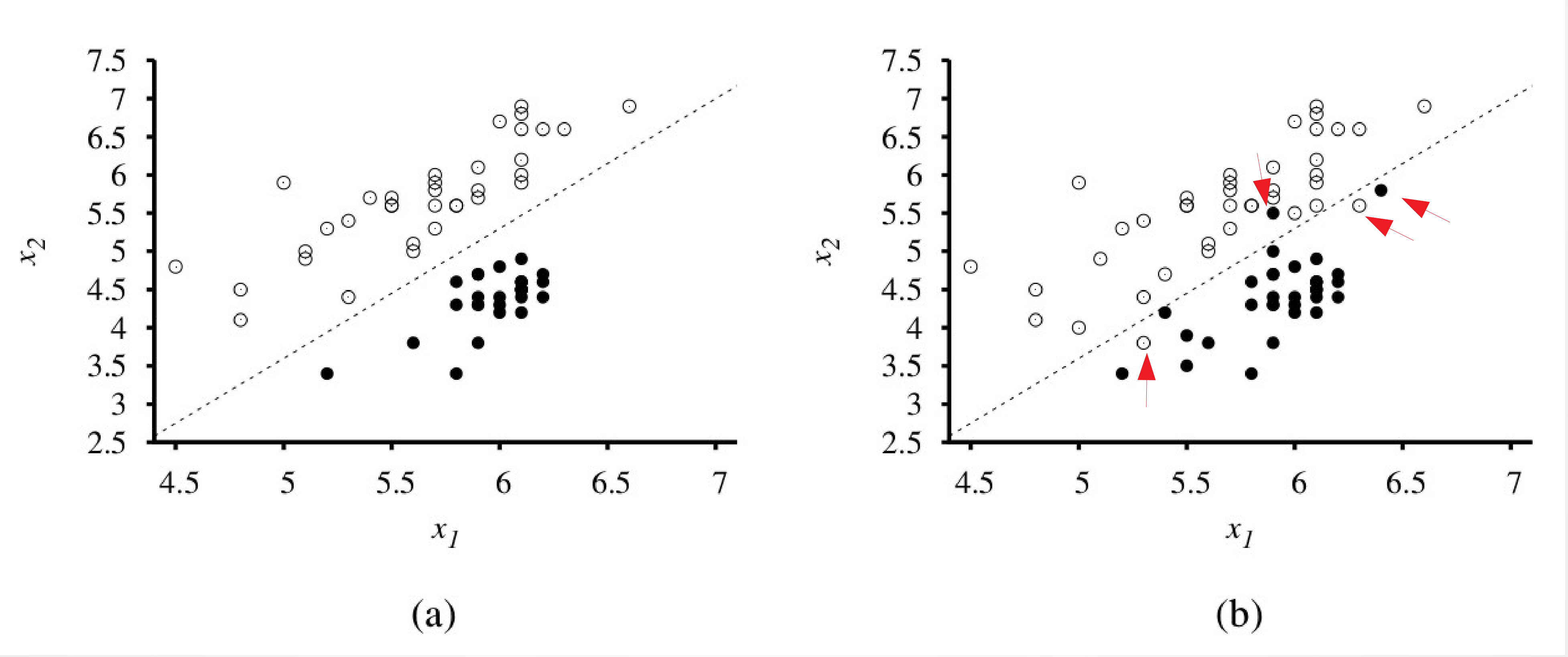
Classificador Linear com Limiar Rígido

- **Classificador Linear**: uso de linha/reta para separar os dados de treinamento



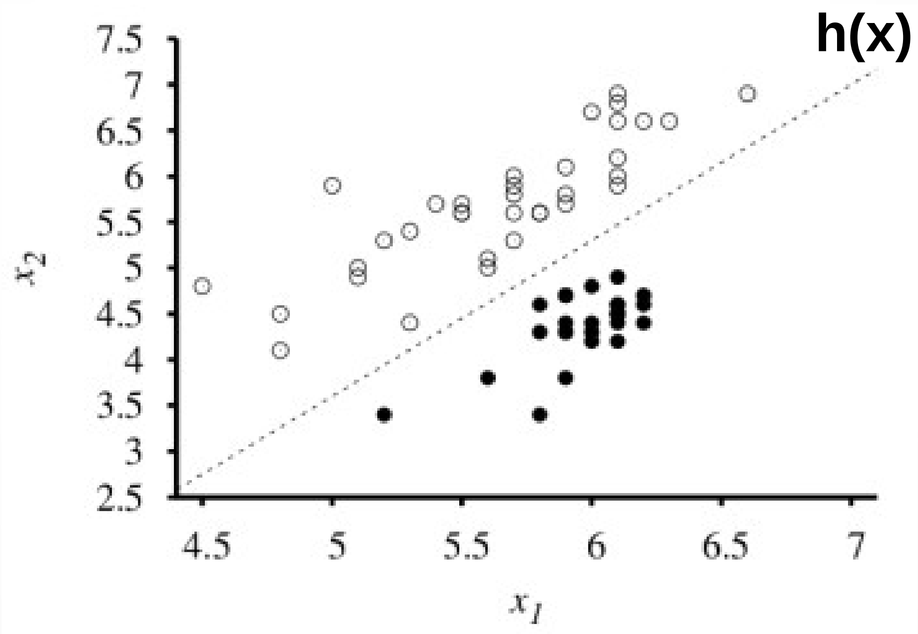
Classificador Linear com Limiar Rígido

- **Classificador Linear**: uso de linha/reta para separar os dados de treinamento



Classificador Linear com Limiar Rígido

- **Classificador Linear:** uso de linha/reta para separar os dados de treinamento



$$h_{\mathbf{w}}(\mathbf{x}) = \mathit{Threshold}(\mathbf{w} \cdot \mathbf{x}) \text{ where } \mathit{Threshold}(z) = 1 \text{ if } z \geq 0 \text{ and } 0 \text{ otherwise.}$$

Classificador Linear com Limiar Rígido

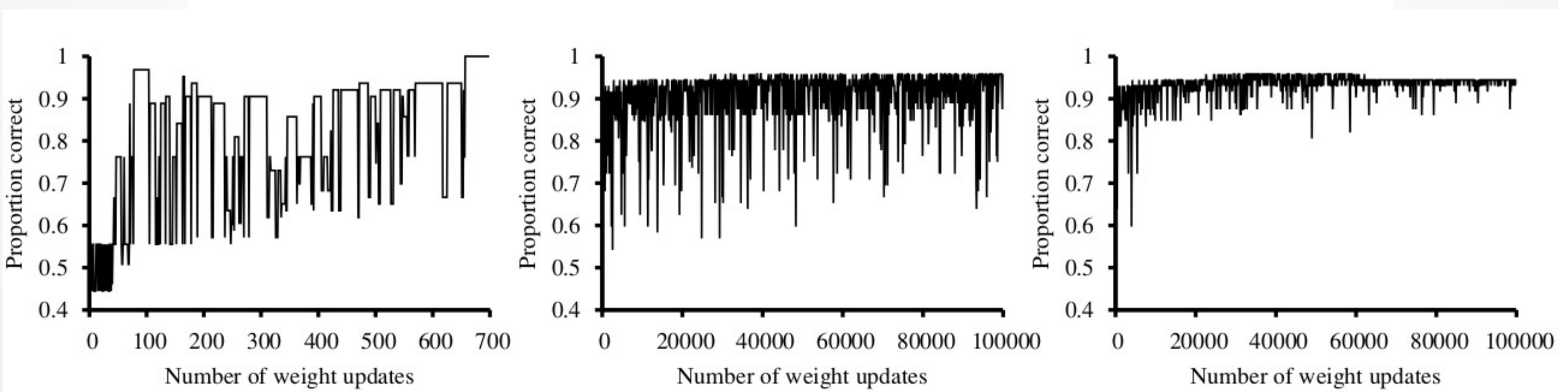
- **<u>Descida pelo Gradiente</u> para atualização dos pesos W={w0,w1}:**

$\mathbf{w} \leftarrow$ any point in the parameter space

loop until convergence **do**

  for each w_i **in w do**

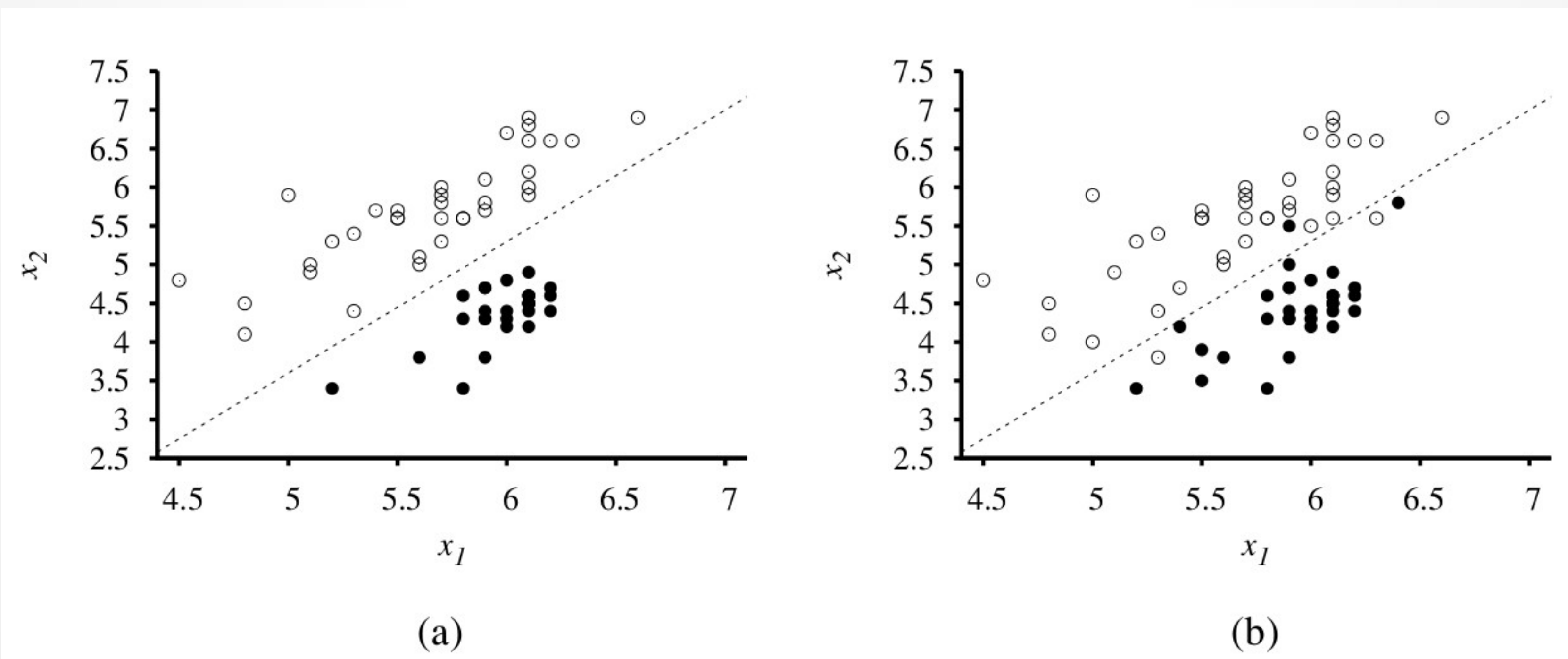
$$w_i \leftarrow w_i + \alpha\,(y - h_{\mathbf{w}}(\mathbf{x})) \times x_i$$



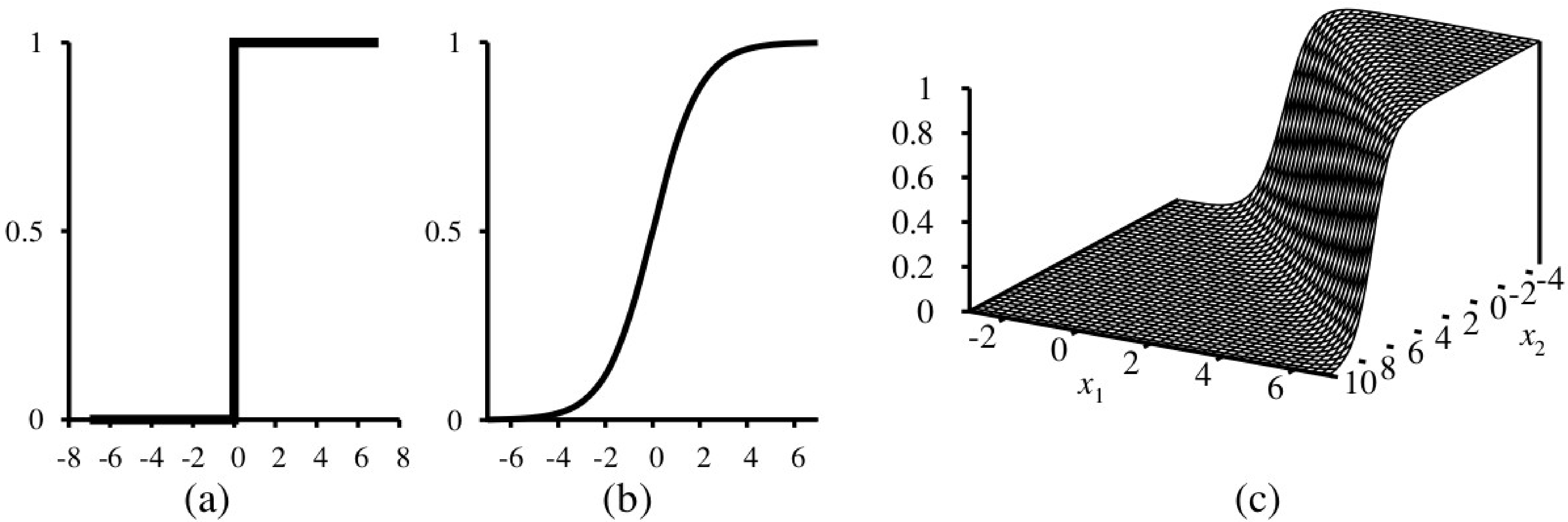
(a) (b) (c) $\alpha(t) = 1000/(1000 + t)$.

Classificador Linear com Regressão Logística

- **Classificador Linear com Regressão Logística:** utiliza de uma funçãoa logística para suavisar a classificação das amostras na <u>fronteira de decisão;</u> $Logistic(z) = \frac{1}{1+e^{-z}}$
- Maior tolerância para amostras em área de risco (fronteira entre classes.



Classificador Linear com Regressão Logística



Logistica(z) é função de transformação

Figure 18.17 (a) The hard threshold function $Threshold(z)$ with 0/1 output. Note that the function is nondifferentiable at $z=0$. (b) The logistic function, $Logistic(z) = \frac{1}{1+e^{-z}}$, also known as the sigmoid function. (c) Plot of a logistic regression hypothesis $h_{\mathbf{w}}(\mathbf{x}) = Logistic(\mathbf{w} \cdot \mathbf{x})$ for the data shown in Figure 18.15(b).

Classificador Linear com Regressão Logística

Logistica(z) é função de transformação

$$h_{\mathbf{w}}(\mathbf{x}) = Logistic(\mathbf{w} \cdot \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}} .$$

chain rule: $\partial g(f(x))/\partial x = g'(f(x))\, \partial f(x)/\partial x$.)

$$\begin{aligned}
\frac{\partial}{\partial w_i} Loss(\mathbf{w}) &= \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x}))^2 \\
&= 2(y - h_{\mathbf{w}}(\mathbf{x})) \times \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x})) \\
&= -2(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w} \cdot \mathbf{x}) \times \frac{\partial}{\partial w_i} \mathbf{w} \cdot \mathbf{x} \\
&= -2(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w} \cdot \mathbf{x}) \times x_i .
\end{aligned}$$

The derivative g' of the logistic function satisfies $g'(z) = g(z)(1 - g(z))$, so we have

$$g'(\mathbf{w} \cdot \mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})(1 - g(\mathbf{w} \cdot \mathbf{x})) = h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x}))$$

Classificador Linear com Regressão Logística

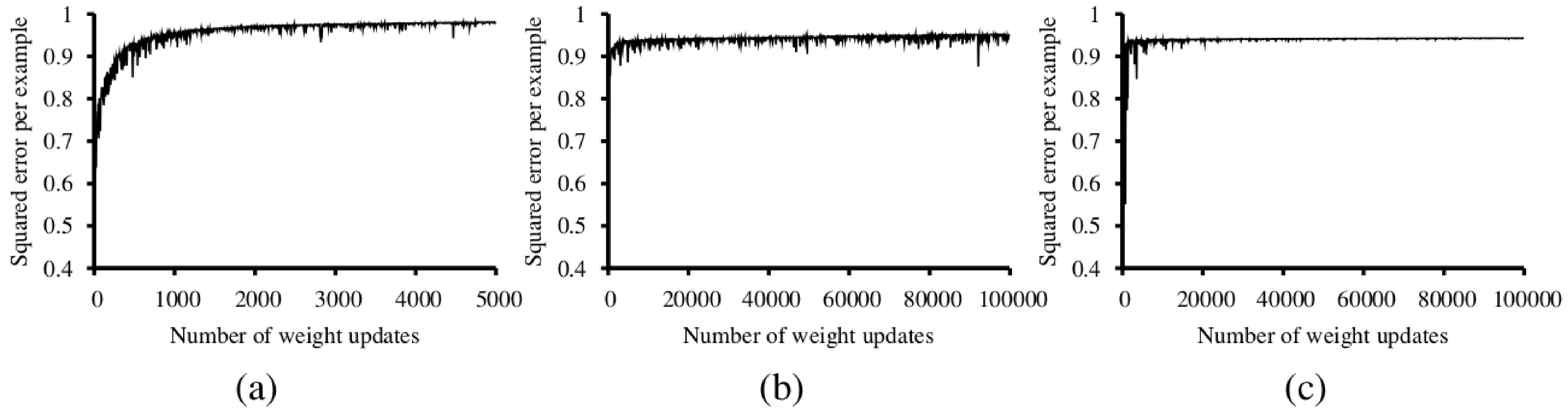


Figure 18.18 Repeat of the experiments in Figure 18.16 using logistic regression and squared error. The plot in (a) covers 5000 iterations rather than 1000, while (b) and (c) use the same scale.

Maior estabilidade durante treinamento do modelo.

Referências

- Peter Norvig e Stuart Russel. Inteligência Artificial. Cap. 18. Seção 3.
- Alguns Slides da Profa. Bianca Zadrozny (IC-UFF)