Regressão Logística

Prof. Danilo Silva

EEL7514/EEL7513 - Tópico Avançado em Processamento de Sinais
EEL410250 - Aprendizado de Máquina

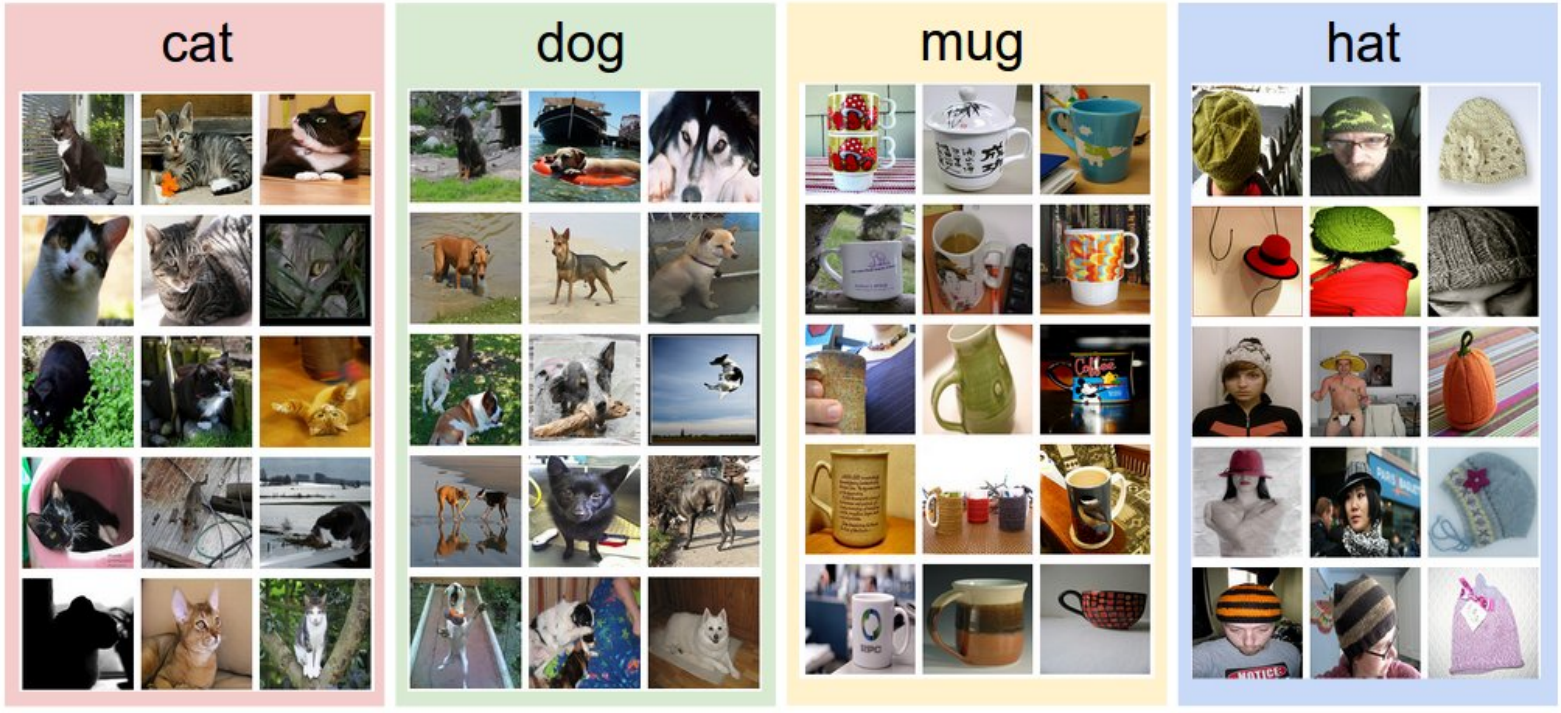
EEL / CTC / UFSC

Tópicos

- Classificação: conceitos gerais
- Classificação binária
- Regressão logística
- Classificação multi-classe
- Avaliação de classificadores binários

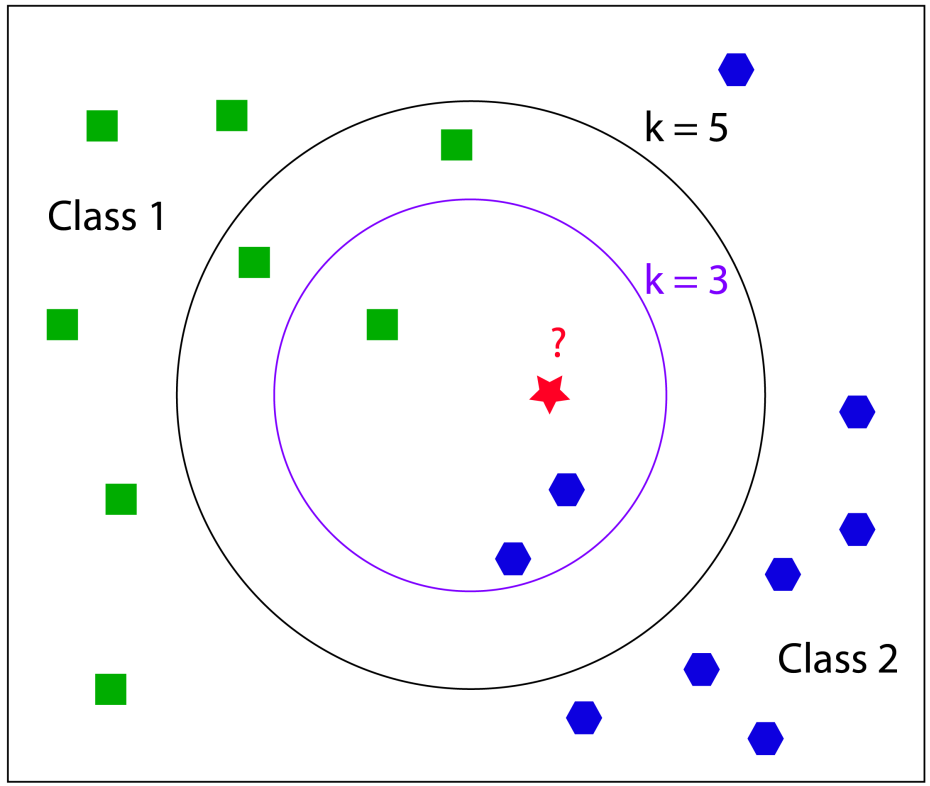
Classificação: Conceitos Gerais

Classificação



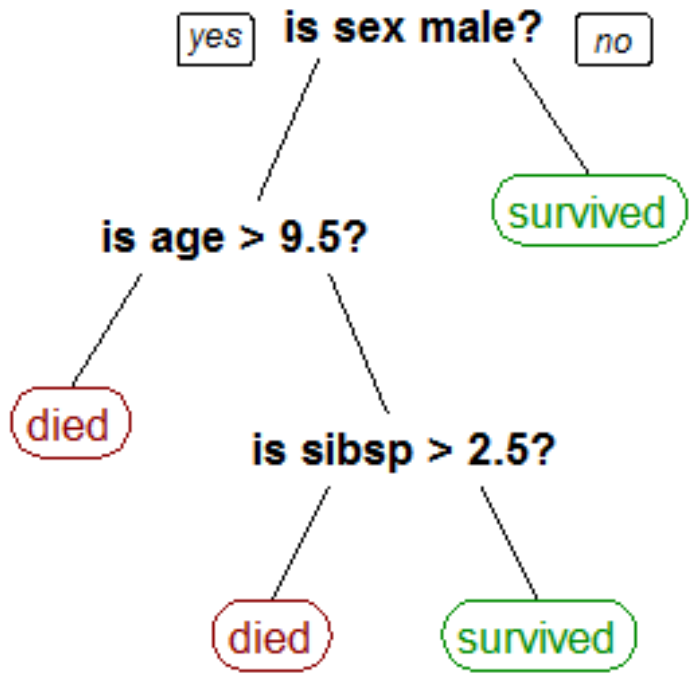
- Problema de classificação com K classes:
  - $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ é o vetor de atributos
  - $y \in \mathcal{Y} = \{1, 2, \ldots, K\}$ é o rótulo que indica a classe a qual $\mathbf{x}$ pertence
  - Um classificador é uma função $\mathbf{x} \mapsto \hat{y} \in \mathcal{Y} = \{1, 2, \ldots, K\}$
- Dado um conjunto de treinamento, desejamos um classificador que consiga prever corretamente a classe de novas amostras

Exemplo: Classificador k-NN (k-*nearest neighbors*)



- Classifica uma nova amostra com a classe mais comum (voto de maioria) entre as dos seus k vizinhos mais próximos no conjunto de treinamento
  - Requer uma medida de distância (ex: distância euclidiana)
  - k é um hiperparâmetro

Exemplo: Árvore de Decisão



$x_1 \in \{0 \text{ (female)}, 1 \text{ (male)}\}$

$x_2 =$ age

$x_3 =$ number of siblings or spouses

$y \in \{0 \text{ (died)}, 1 \text{ (survived)}\}$

Classificação $\times$ Regressão

- Os rótulos das classes $\mathcal{Y} = \{1, \ldots, K\}$ correspondem a um mapeamento arbitrário de algum conjunto
  - Ex: $\{1 \to \text{cat},\ 2 \to \text{dog},\ 3 \to \text{mug},\ 4 \to \text{hat}\}$
  - Ex: $\{3 \to \text{cat},\ 1 \to \text{dog},\ 4 \to \text{mug},\ 2 \to \text{hat}\}$
- Não existe uma ordenação natural entre as classes
- No jargão da área, este tipo de variável é conhecida como categórica, em oposição a uma variável numérica $y \in \mathbb{R}$ (valor-alvo de regressão)
- O que fazer se $y \in \{1, \ldots, K\}$ permite uma interpretação numérica?
  - Ex: avaliação de um produto: 1 a 5 estrelas
- Nesse caso, o problema pode ser formulado tanto como classificação quanto como regressão—mas tipicamente é mais fácil resolver como regressão

Desempenho de um Classificador

- Considere o conjunto de dados $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$ e seja $\hat{y}^{(i)}$ a predição do classificador sobre a amostra $(\mathbf{x}^{(i)}, y^{(i)})$.

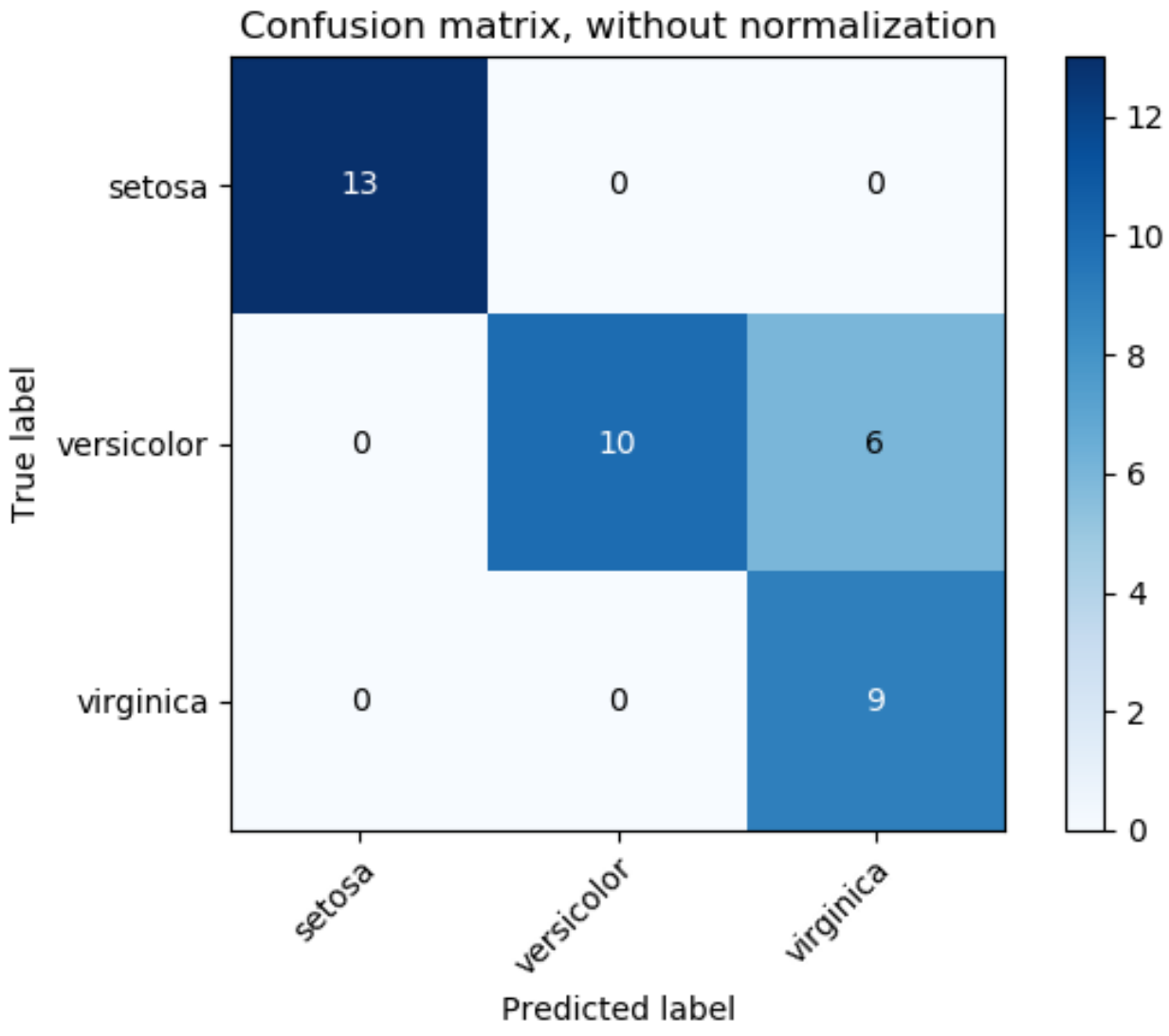
- A forma mais geral de avaliar o desempenho de um classificador sobre um conjunto de dados é através da sua matriz de confusão

$$\mathbf{N} = \begin{bmatrix} N(1,1) & \cdots & N(1,K) \\ \vdots & \ldots & \vdots \\ N(K,1) & \cdots & N(K,K) \end{bmatrix}$$

  onde $N(y, \hat{y})$ denota o número de amostras de $\mathcal{D}$ que pertencem à classe y e foram classificadas como $\hat{y}$.

- No entanto, esta é uma avaliação multi-objetivo; na prática, é útil sumarizar o desempenho em uma métrica de um único número (*single-real-number evaluation metric*)

Exemplo: Matriz de Confusão



Desempenho de um Classificador

- Uma forma simples e muito utilizada de avaliar o desempenho de um classificador é através da perda média

$$J = \frac{1}{m} \sum_{i=1}^{m} L(y^{(i)}, \hat{y}^{(i)})$$

onde $L(y, \hat{y})$ denota a perda ou custo de se classificar uma amostra como $\hat{y}$ quando a classe correta é y

- Nesse caso, temos

$$J = \frac{1}{m} \sum_{y=1}^{K} \sum_{\hat{y}=1}^{K} N(y, \hat{y}) L(y, \hat{y})$$

Exemplo

- Perda 0-1 (*zero-one loss*)

$$L(y, \hat{y}) = 1[y \neq \hat{y}] = \begin{cases} 0, & y = \hat{y} \\ 1, & y \neq \hat{y} \end{cases}$$

onde $1[\cdot]$ é uma função indicadora dada por

$$1[P] = \begin{cases} 1, & \text{se } P \text{ é verdadeira} \\ 0, & \text{se } P \text{ é falsa} \end{cases}$$

  - Todo acerto tem custo zero, todo erro tem o mesmo custo
- Nesse caso, a perda média corresponde à taxa de erro
- Acurácia = $1 -$ taxa de erro

Regiões de Decisão

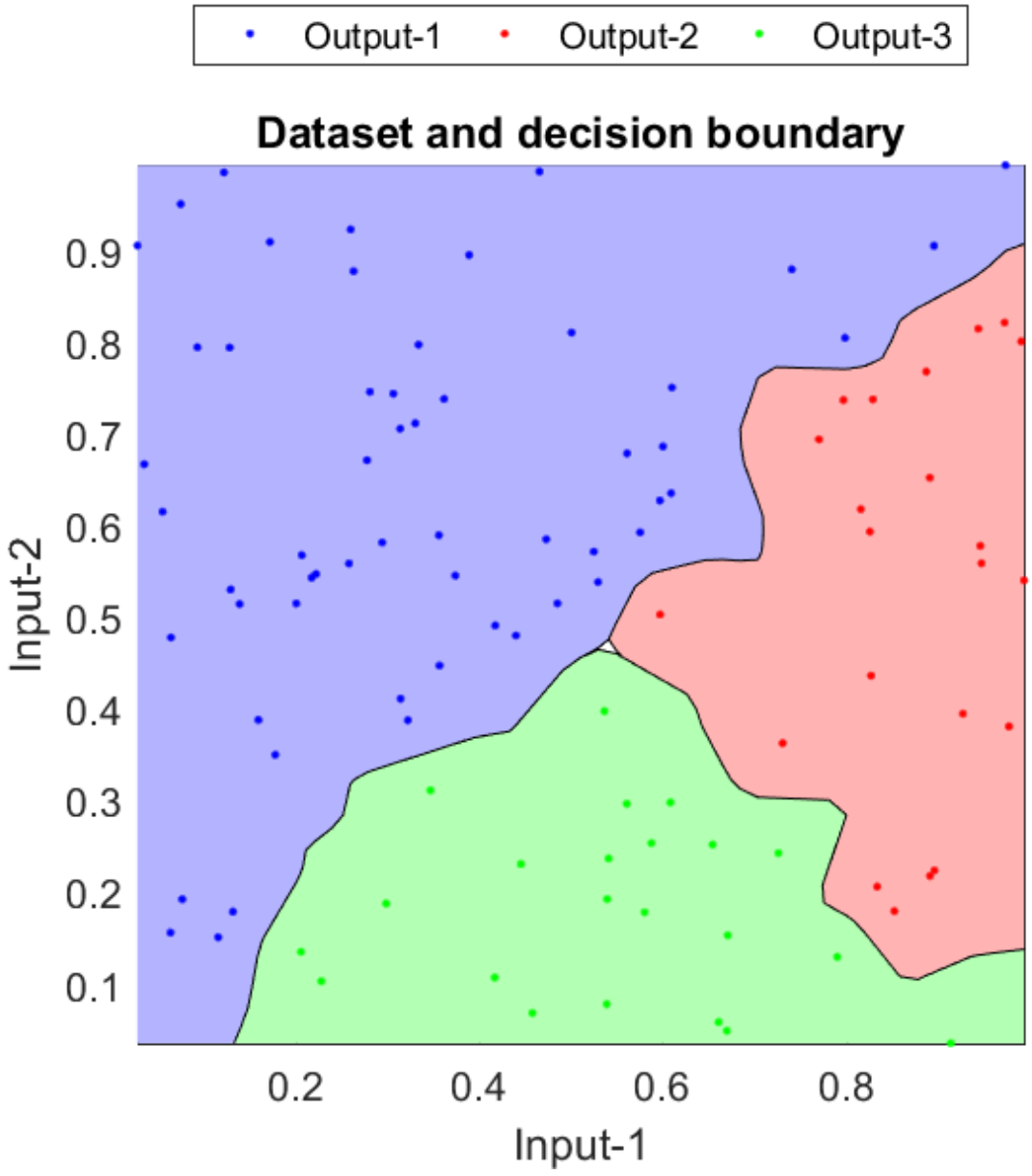
- Em geral, um classificador pode ser equivalentemente representado por uma partição de $\mathbb{R}^n$ em regiões de decisão $\mathcal{R}_1, \ldots, \mathcal{R}_K$, de tal forma que

$$\hat{y} = \begin{cases} 1, & \text{se } \mathbf{x} \in \mathcal{R}_1 \\ \vdots & \vdots \\ K, & \text{se } \mathbf{x} \in \mathcal{R}_K \end{cases} \quad = \quad \sum_{k=1}^{K} k \cdot 1[\mathbf{x} \in \mathcal{R}_k]$$

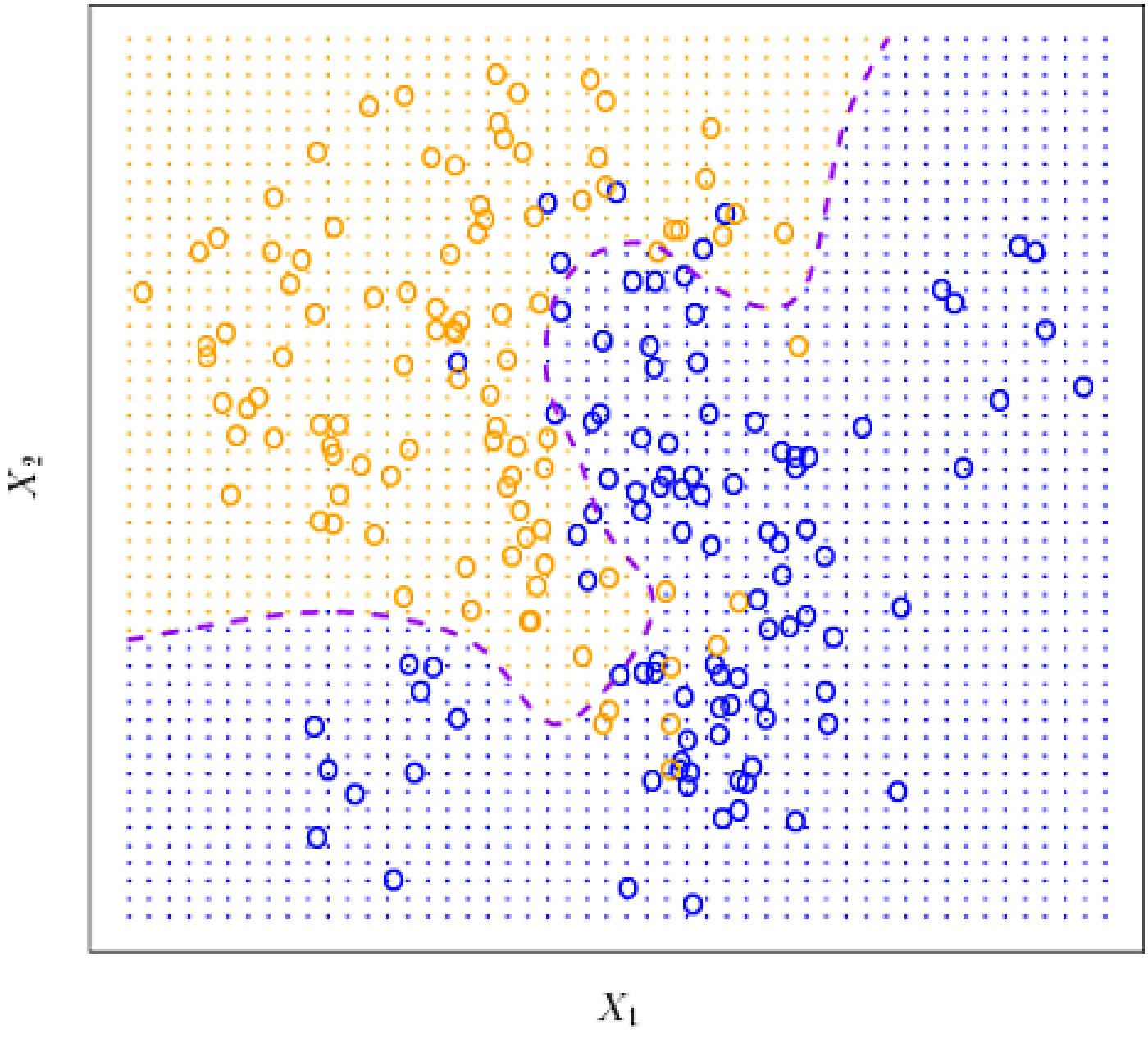
onde $1[\cdot]$ é uma função indicadora dada por

$$1[P] = \begin{cases} 1, & \text{se } P \text{ é verdadeira} \\ 0, & \text{se } P \text{ é falsa} \end{cases}$$

Exemplo


Output-1
Output-2
Output-3
Dataset and decision boundary
Input-2
Input-1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.4
0.6
0.8

Exemplo



Funções Discriminantes

- Sem perda de generalidade[1], podemos representar um classificador através de funções discriminantes $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}) \in \mathbb{R}$
  - Ex: $f_k(\mathbf{x}) = 1[\mathbf{x} \in \mathcal{R}_k]$
- Decide-se pela classe k que maximiza o discriminante:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; f_k(\mathbf{x})$$

- Informalmente, podemos interpretar $f_k(\mathbf{x})$ como um grau (ou *score*) de confiança de que a amostra $\mathbf{x}$ percence à classe k
- Assim, o problema de classificação é transformado em K problemas de regressão

[1] Obs: embora não haja perda de generalidade nessa representação, nem todo classificador de fato a utiliza na prática. Ex: árvores de decisão, classificador k-nn

Classificação Linear

- O tipo mais simples de discriminante é o discriminante linear

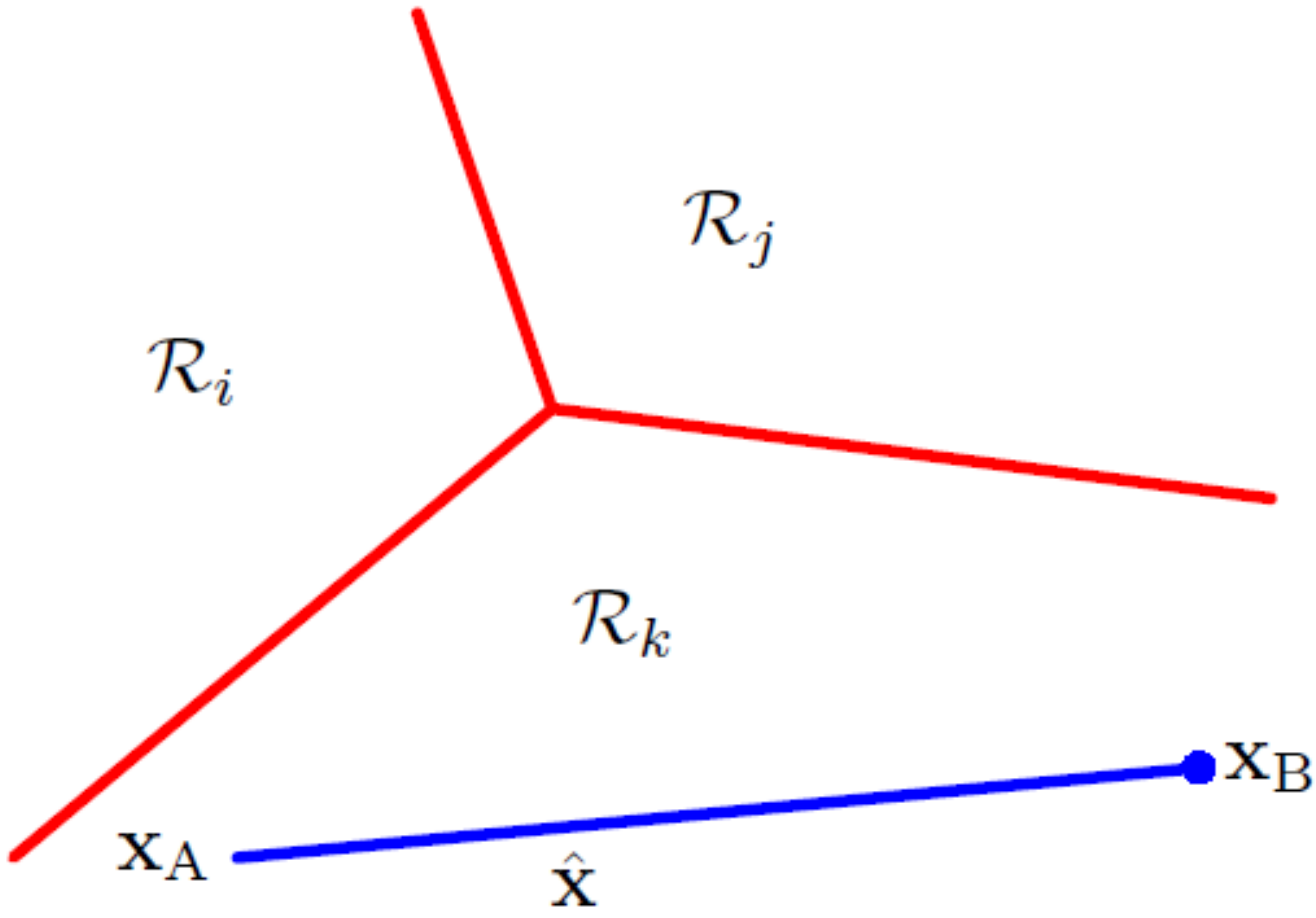
$$f_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + b_k, \qquad k = 1, \ldots, K$$

  onde $\mathbf{w}_k = (w_{k,1}, \ldots, w_{k,n})^T$

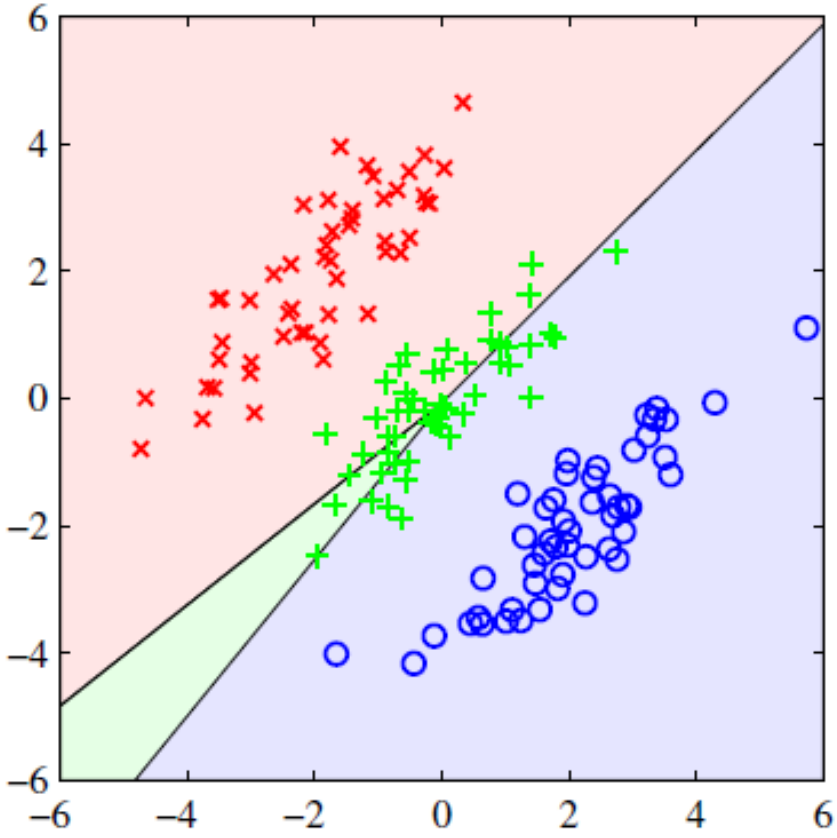
- Um classificador que utiliza discriminantes lineares é chamado de classificador linear

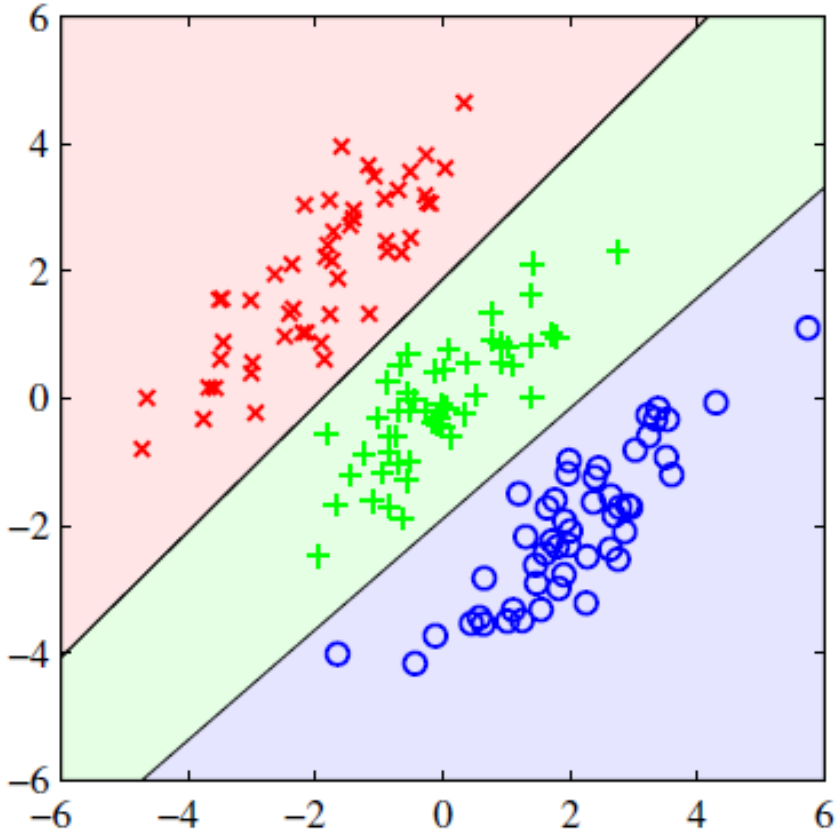
- Nesse caso, as regiões de decisão são separadas através de hiperplanos em $\mathbb{R}^n$
  - $\Longrightarrow$ regiões “simplesmente conexas” e convexas

Exemplo

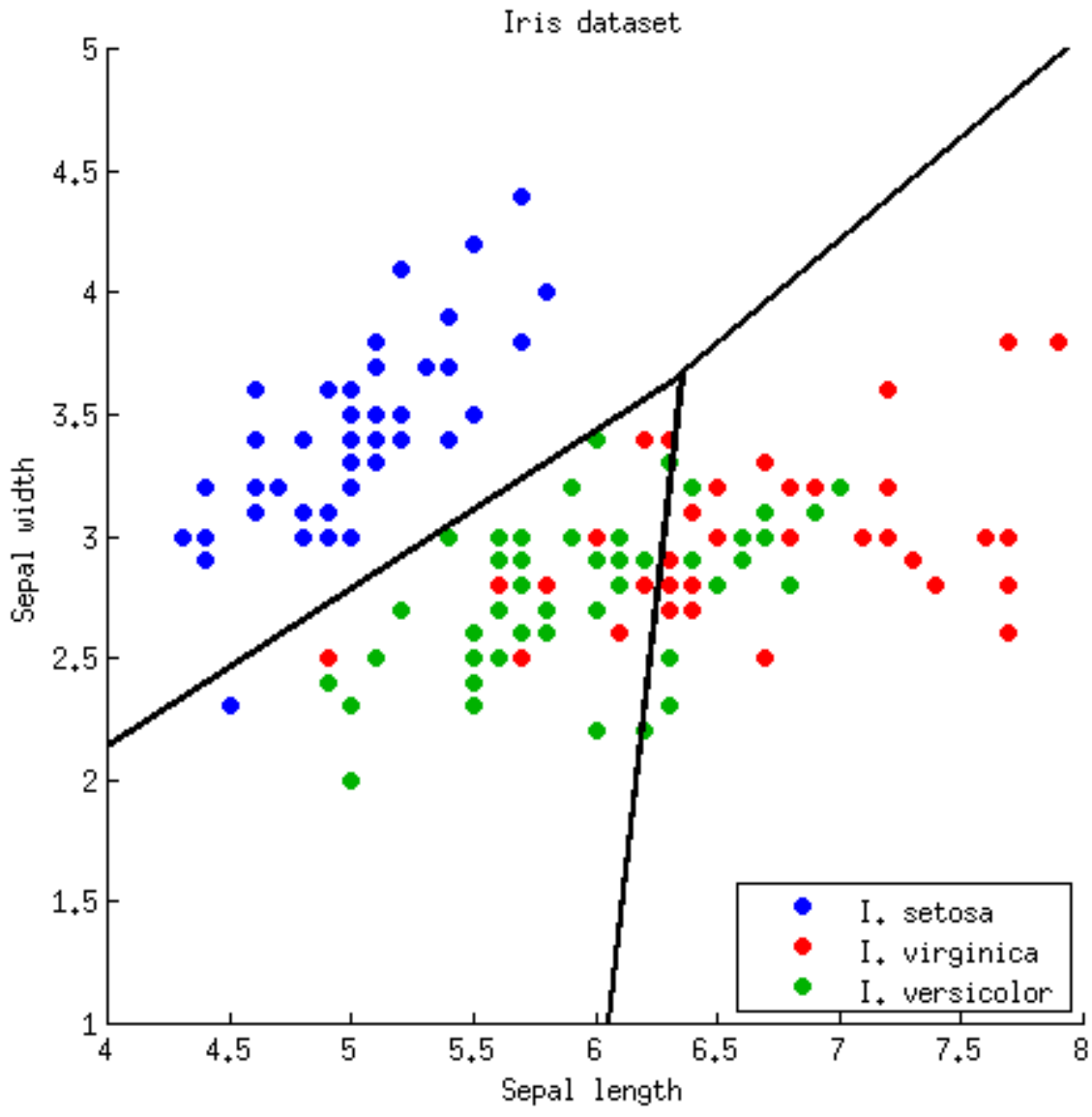

$\mathcal{R}_j$
$\mathcal{R}_i$
$\mathcal{R}_k$
$\mathbf{x}_\mathrm{B}$
$\mathbf{x}_\mathrm{A}$
$\hat{\mathbf{x}}$

Exemplo





Exemplo


Iris dataset
Sepal width
Sepal length
5
4.5
4
3.5
3
2.5
2
1.5
1
4
4.5
5
5.5
6
6.5
7
7.5
8
I. setosa
I. virginica
I. versicolor

Classificação Binária

Notação

- Se $K = 2$, temos um problema de classificação binária
- Nesse caso, ao invés de considerar o conjunto de rótulos $\mathcal{Y} = \{1, 2\}$, é mais conveniente e mais comum considerar
  - $\mathcal{Y} = \{0$ (classe negativa), 1 (classe positiva)$\}$
  - $\mathcal{Y} = \{-1$ (classe negativa), $+1$ (classe positiva)$\}$

  (Na literatura, é comum inclusive alternar entre as duas notações várias vezes ao longo de um texto)
- Para manter a consistência da notação, usaremos sempre:
  - $y, \hat{y} \in \mathcal{Y} = \{0, 1\}$
  - $y_s, \hat{y}_s \in \{-1, +1\}$, i.e., $y_s = 2y - 1$ e $y = (y_s + 1)/2$

Classificação Binária

- Na classificação binária, é suficiente usar um único discriminante:

$$\begin{aligned} \hat{y} = 1 &\iff f_1(\mathbf{x}) > f_0(\mathbf{x}) \\ &\iff f(\mathbf{x}) \triangleq f_1(\mathbf{x}) - f_0(\mathbf{x}) > 0 \end{aligned}$$

  (consequentemente, $\hat{y} = 0$ se $f(\mathbf{x}) < 0$)

- De forma mais compacta, podemos escrever:

$$\hat{y} = 1[f(\mathbf{x}) > 0]$$

Classificação Binária Linear

- No caso de um classificador linear, temos

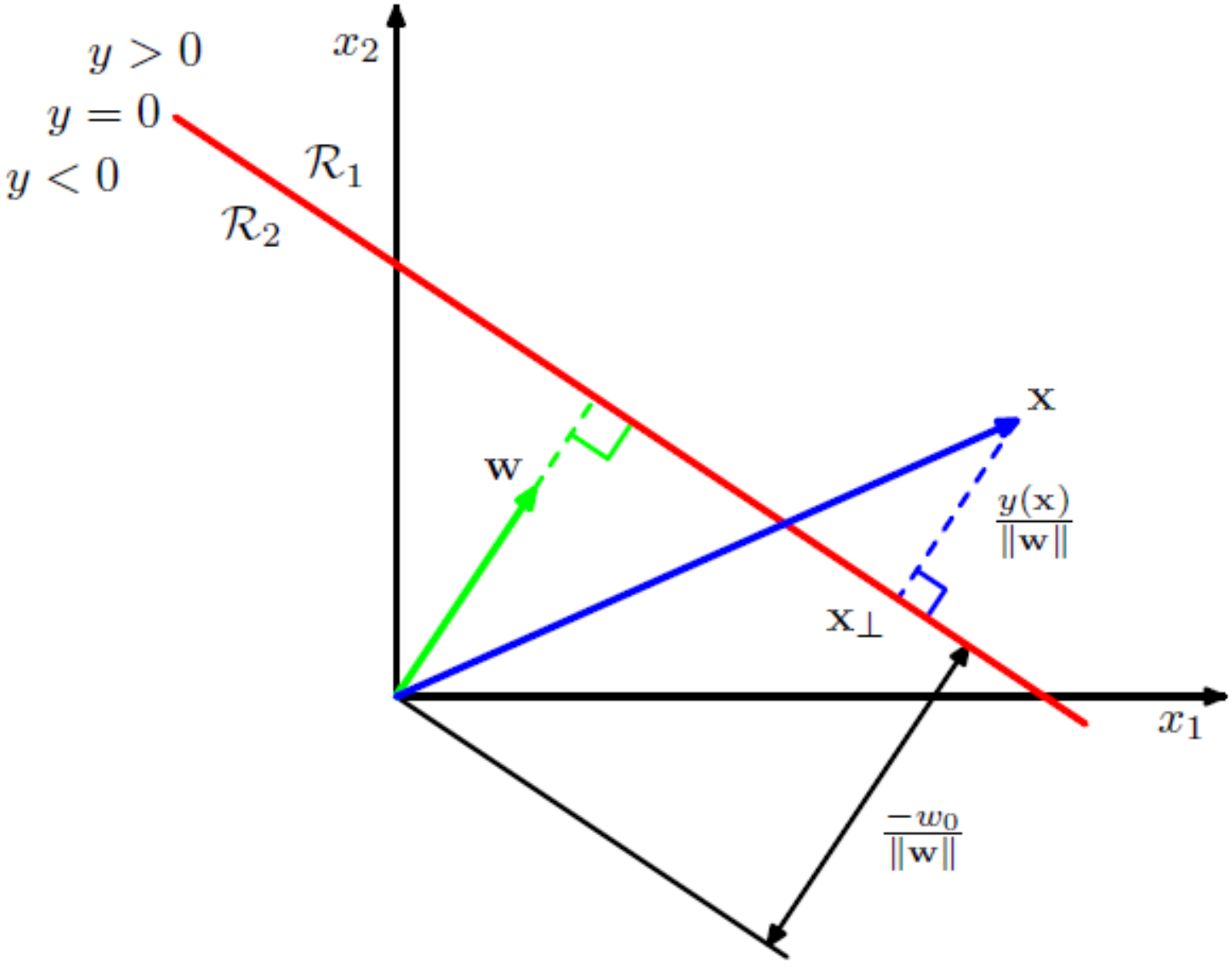
$$\hat{y} = 1 \iff \mathbf{w}^T\mathbf{x} + b > 0$$

onde $\mathbf{w} = (w_1, \ldots, w_n)^T$, ou simplesmente

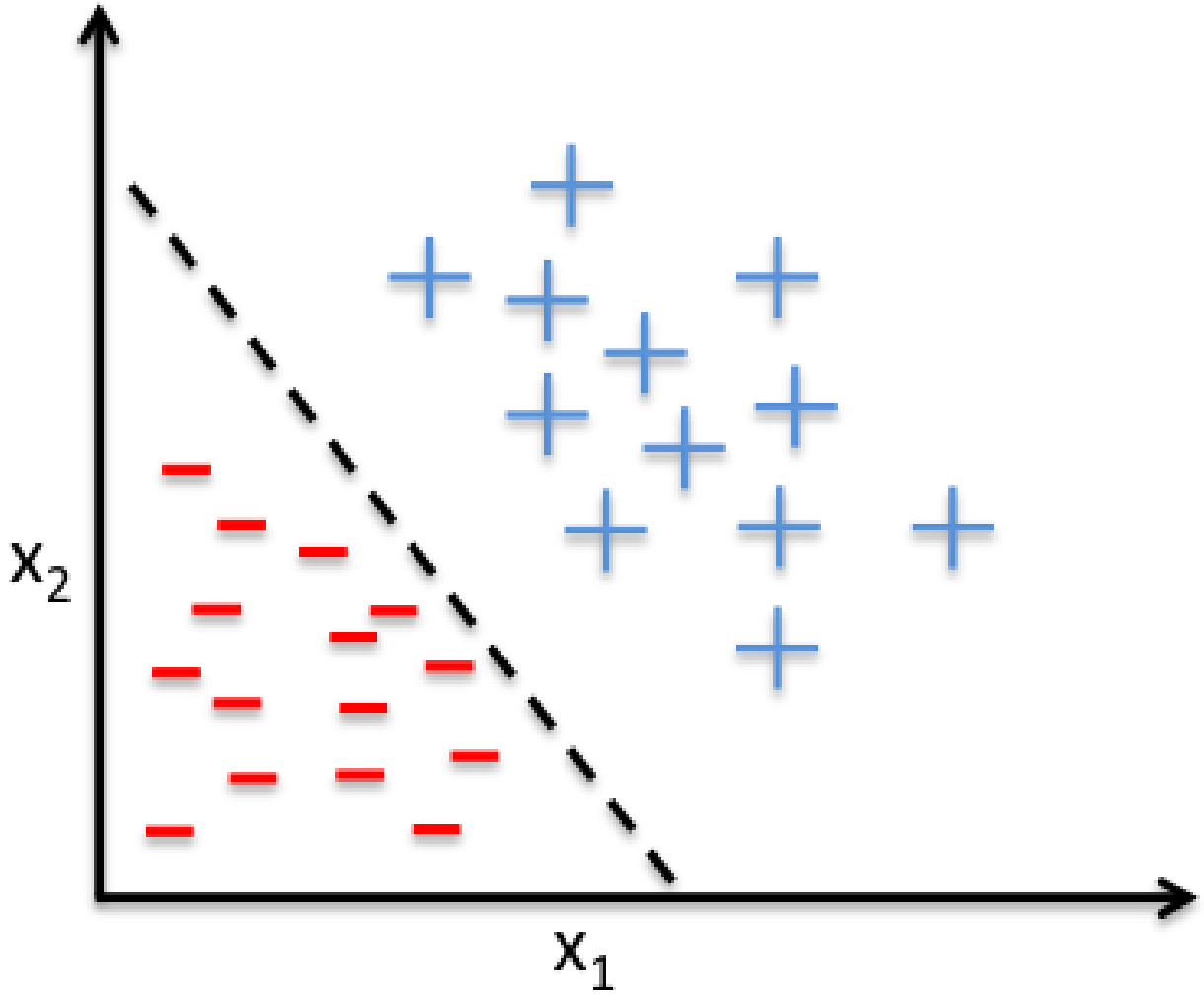
$$\hat{y} = 1[\mathbf{w}^T\mathbf{x} + b > 0]$$

- Geometricamente, a equação $\mathbf{w}^T\mathbf{x} + b = 0$ define um hiperplano em $\mathbb{R}^n$ perpendicular a $\mathbf{w}$ e que passa pelo ponto $-\frac{b}{\|\mathbf{w}\|}\frac{\mathbf{w}}{\|\mathbf{w}\|}$
- Uma amostra $\mathbf{x}$ é classificada como positiva ($\hat{y} = 1$) se estiver no semi-espaço do lado positivo do hiperplano (no sentido da projeção na direção de $\mathbf{w}$), caso contrário é classificada como negativa ($\hat{y} = 0$)

Exemplo

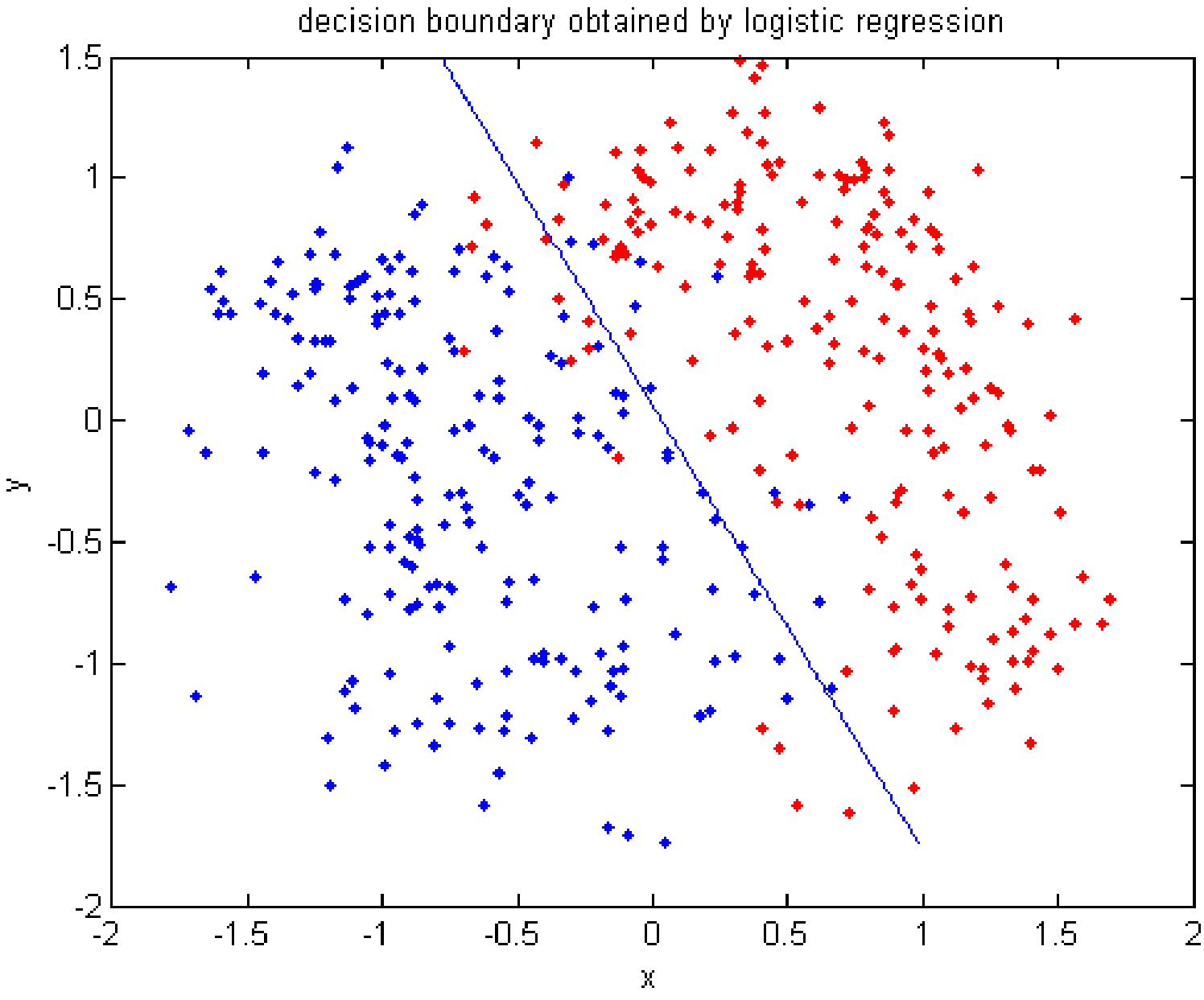

$y > 0$
$y = 0$
$y < 0$
x_2
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathbf{x}$
$\mathbf{w}$
$\frac{y(\mathbf{x})}{\|\mathbf{w}\|}$
$\mathbf{x}_\perp$
x_1
$\frac{-w_0}{\|\mathbf{w}\|}$

Exemplo



Example of a linear decision boundary for binary classification.

Exemplo


decision boundary obtained by logistic regression
1.5
1
0.5
0
-0.5
-1
-1.5
-2
y
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
x

Notação

- Para facilitar, vamos a partir de agora considerar a notação

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

onde $\mathbf{w} = (b, w_1, \ldots, w_n)^T$ e $\mathbf{x} = (1, x_1, \ldots, x_n)^T$

- Assim, a predição será dada simplesmente por

$$\hat{y} = 1[\mathbf{w}^T \mathbf{x} > 0]$$

Classificação Binária via Regressão Linear

- Uma forma simples de determinar $\mathbf{w}$ é usando regressão linear com perda quadrática (i.e., mínimos quadrados)
- Nesse caso, ajustamos um modelo

$$z = \mathbf{w}^T \mathbf{x}$$

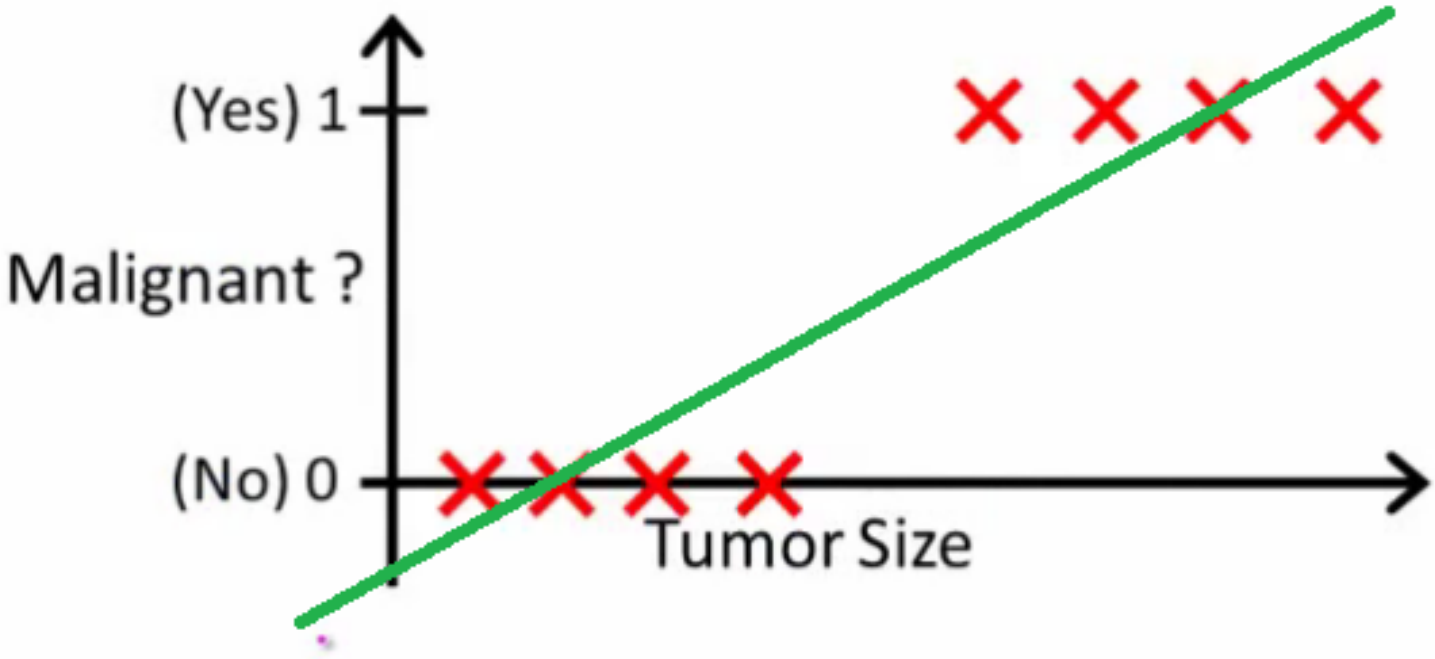
a partir de exemplos de treinamento rotulados como $y_s \in \{-1, +1\}$, utilizando a função perda

$$L(y_s, z) = (y_s - z)^2$$

- Note que a classificação continua sendo dada por

$$\hat{y} = 1[\mathbf{w}^T \mathbf{x} > 0] = 1[z > 0]$$

Exemplo



Problema: Sensibilidade a Outliers

- Um problema desta solução é que o uso do erro quadrático

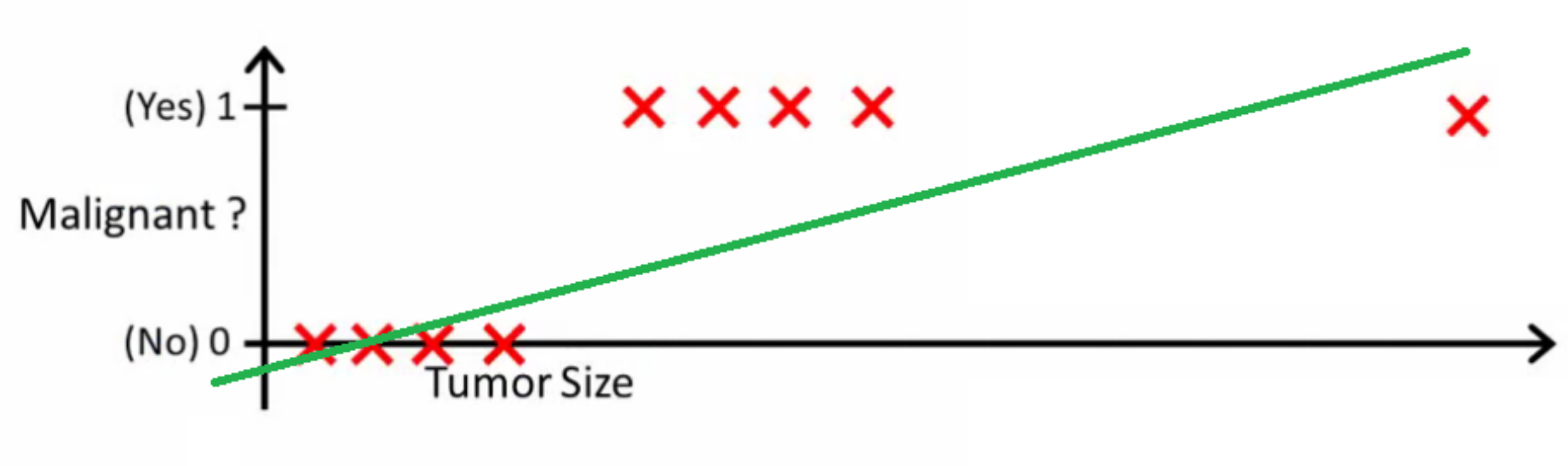
$$L(y_s, z) = (y_s - z)^2$$

  penaliza predições que estão “certas demais”

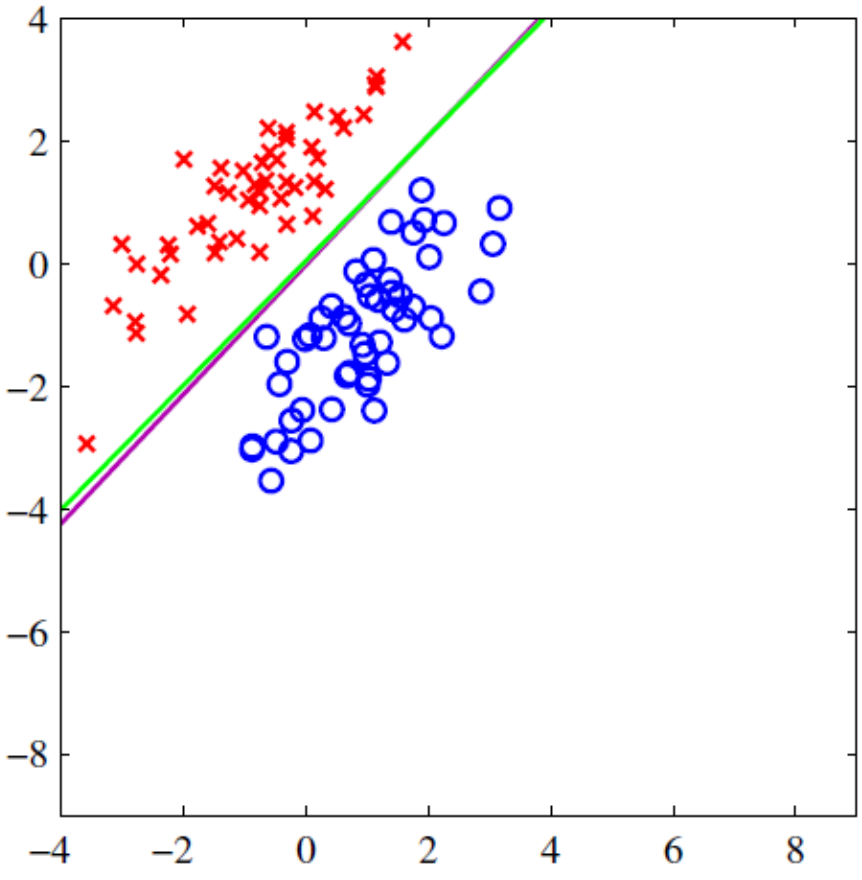
- Por exemplo, supondo $y = y_s = 1$:
  - $z = 100$ (acerto com alta confiança) $\implies L(y_s, z) = 99^2 = 9801$
  - $z = -1$ (erro) $\implies L(y, z) = 4$

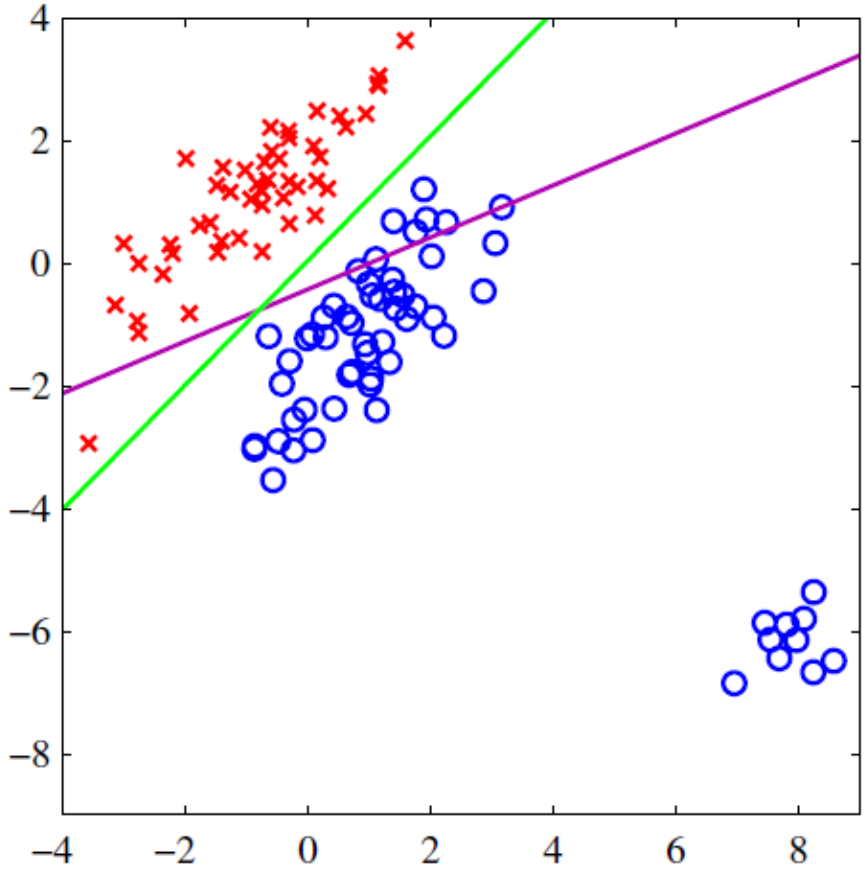
- Consequentemente, valores altos de $z = \mathbf{w}^T\mathbf{x}$ influenciam excessivamente o modelo

Exemplo



Exemplo





Regressão Logística

Regressão Logística

- Uma solução para esse problema é o modelo de regressão logística

$$\tilde{y} = \sigma(\mathbf{w}^T \mathbf{x})$$

com rótulos codificados como $y \in \{0, 1\}$, onde

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

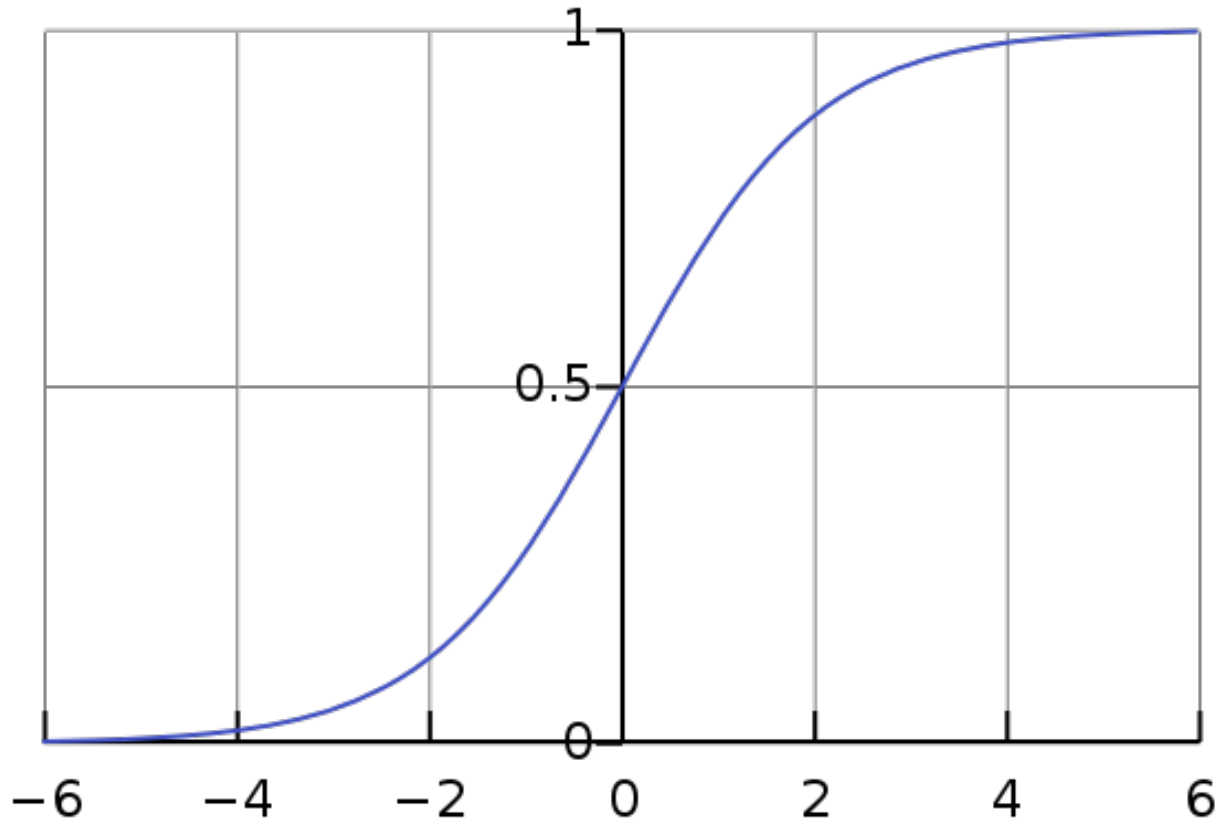
é a função sigmóide logística padrão

- Note que

$$\lim_{z \to -\infty} \sigma(z) = 0, \qquad \sigma(0) = \frac{1}{2}, \qquad \lim_{z \to \infty} \sigma(z) = 1$$

- Classificação: $\hat{y} = 1[\mathbf{w}^T \mathbf{x} > 0] = 1[\tilde{y} > 1/2]$
- Obs: $\tilde{y} \in [0, 1]$ pode ser interpretado como estimativa de $p(y = 1|\mathbf{x})$

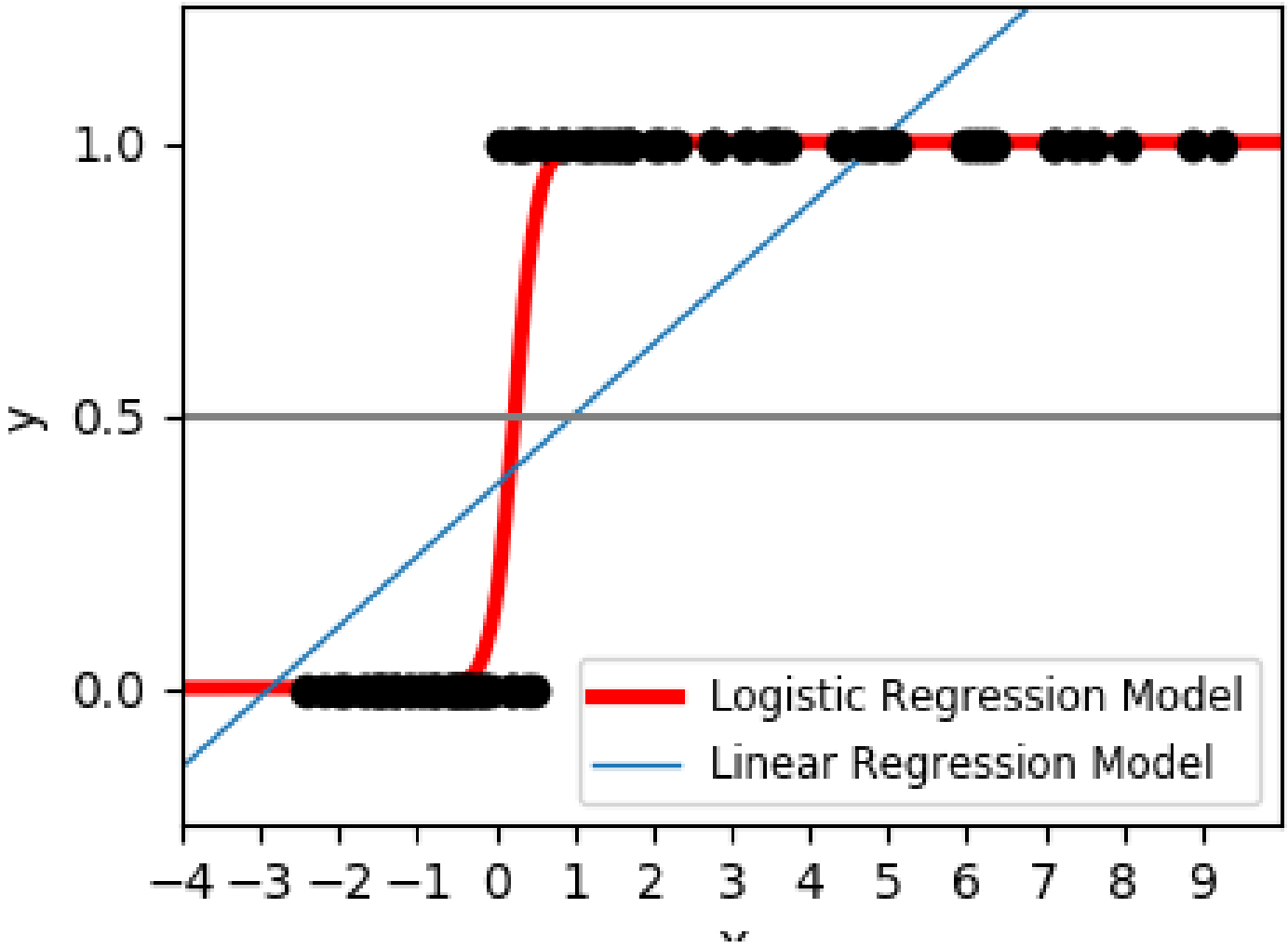
Função Logística



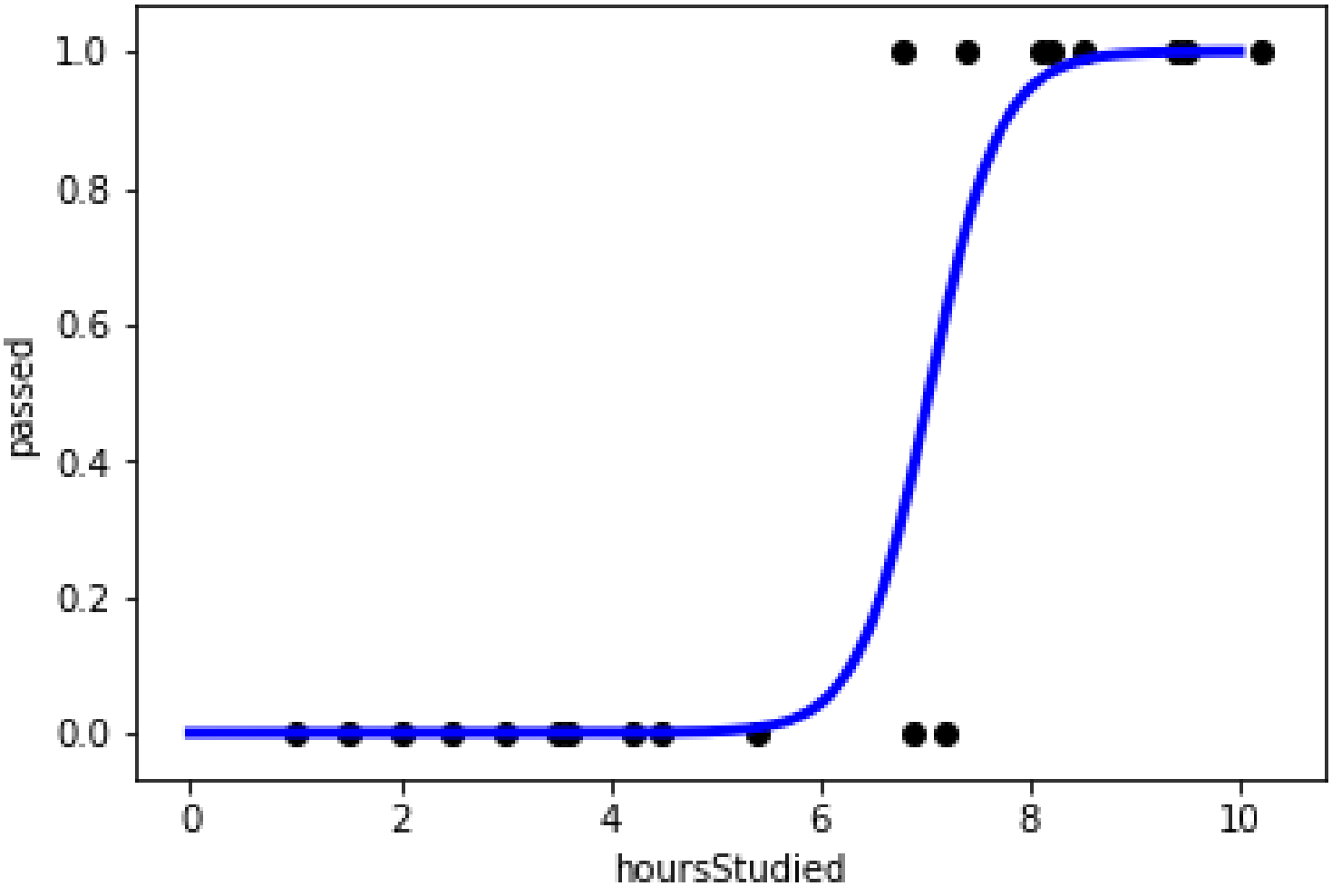
Propriedades:

$$\sigma(-x) = 1 - \sigma(x)$$
$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Exemplo


1.0
0.5
0.0
y
−4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9
x
Logistic Regression Model
Linear Regression Model

Exemplo



Função Perda

- A perda média do modelo é dada por

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

onde $\tilde{y}^{(i)} = \sigma(\mathbf{w}^T \mathbf{x}^{(i)})$

- Mesmo com o modelo de regressão logística, o uso do erro quadrático

$$L(y, \tilde{y}) = (y - \tilde{y})^2$$

ainda é problemático:

  - Penaliza pouco um score de confiança $z = \mathbf{w}^T \mathbf{x}$ muito errado:
    Ex: $y = 1,\ z = -\infty \implies \tilde{y} = 0$ e $L(y, \tilde{y}) = 1$
  - Resulta em uma função custo $J(\mathbf{w})$ não-convexa

Função Perda

- É usual adotar como função perda a entropia cruzada:

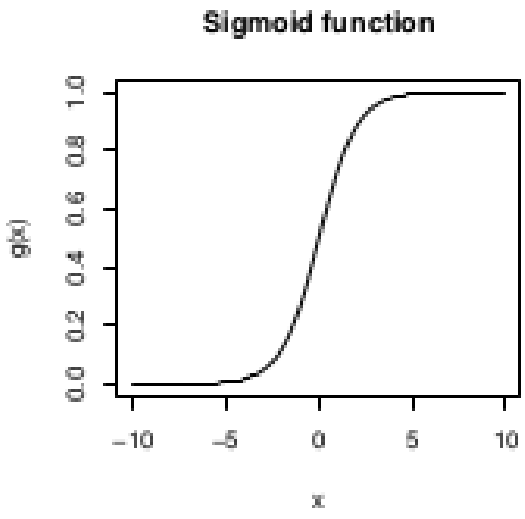
$$L(y, \tilde{y}) = -y \log \tilde{y} - (1 - y) \log(1 - \tilde{y})$$

  - Note que $L(0, 1) = L(1, 0) = \infty$, enquanto $L(0, 0) = L(1, 1) = 0$
  - Resulta em uma função custo $J(\mathbf{w})$ convexa

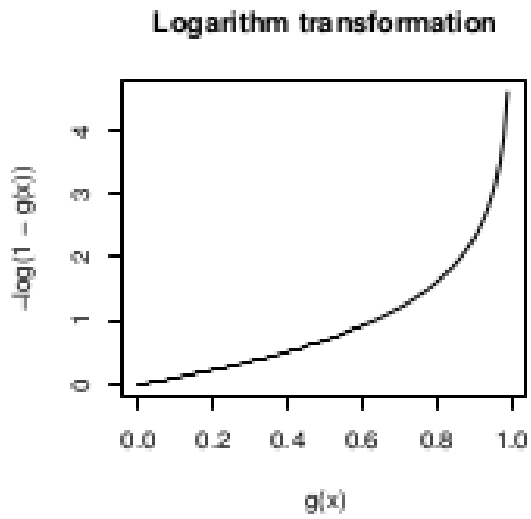
$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (-1) y^{(i)} \log \sigma(\mathbf{w}^T \mathbf{x}^{(i)}) - (1 - y^{(i)}) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}^{(i)}))$$

- Exercício (opcional): prove que $J(\mathbf{w})$ é convexa

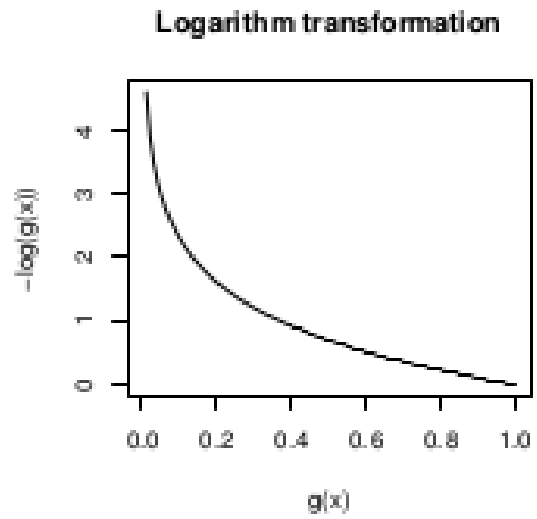
Função Custo: Exemplo



(a) *Sigmoid function.*



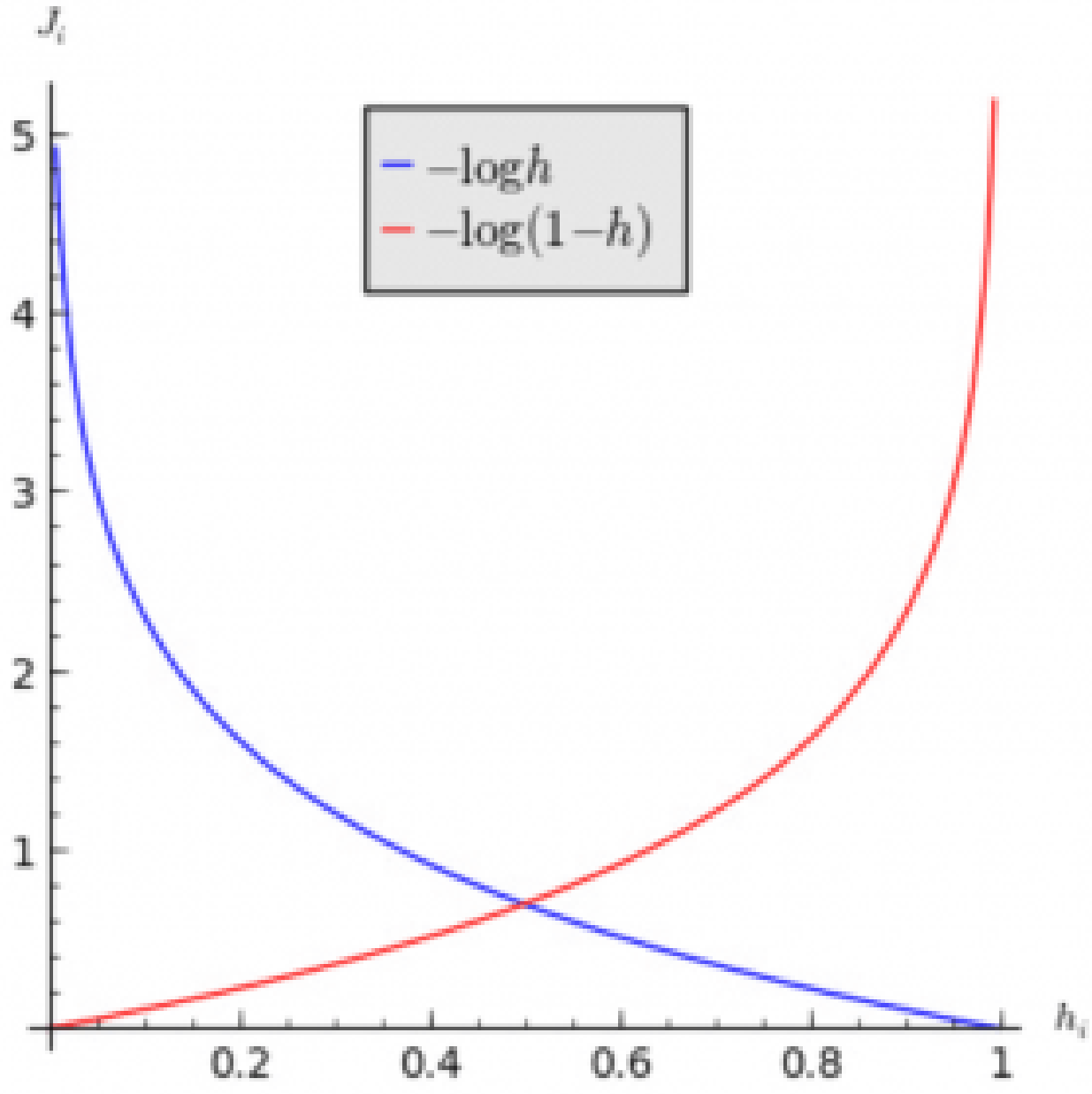
(b) *Cost for* $y = 0$.



(c) *Cost for* $y = 1$.

Figure B.1: *Logarithmic transformation of the sigmoid function.*

Função Custo: Exemplo



Treinamento

- Função custo (em notação vetorial):

$$J(\mathbf{w}) = \frac{1}{m}\left(-\mathbf{y}^T \log \tilde{\mathbf{y}} - (1-\mathbf{y})^T \log(1-\tilde{\mathbf{y}})\right)$$

  onde $\tilde{\mathbf{y}} = \sigma(\mathbf{X}\mathbf{w})$

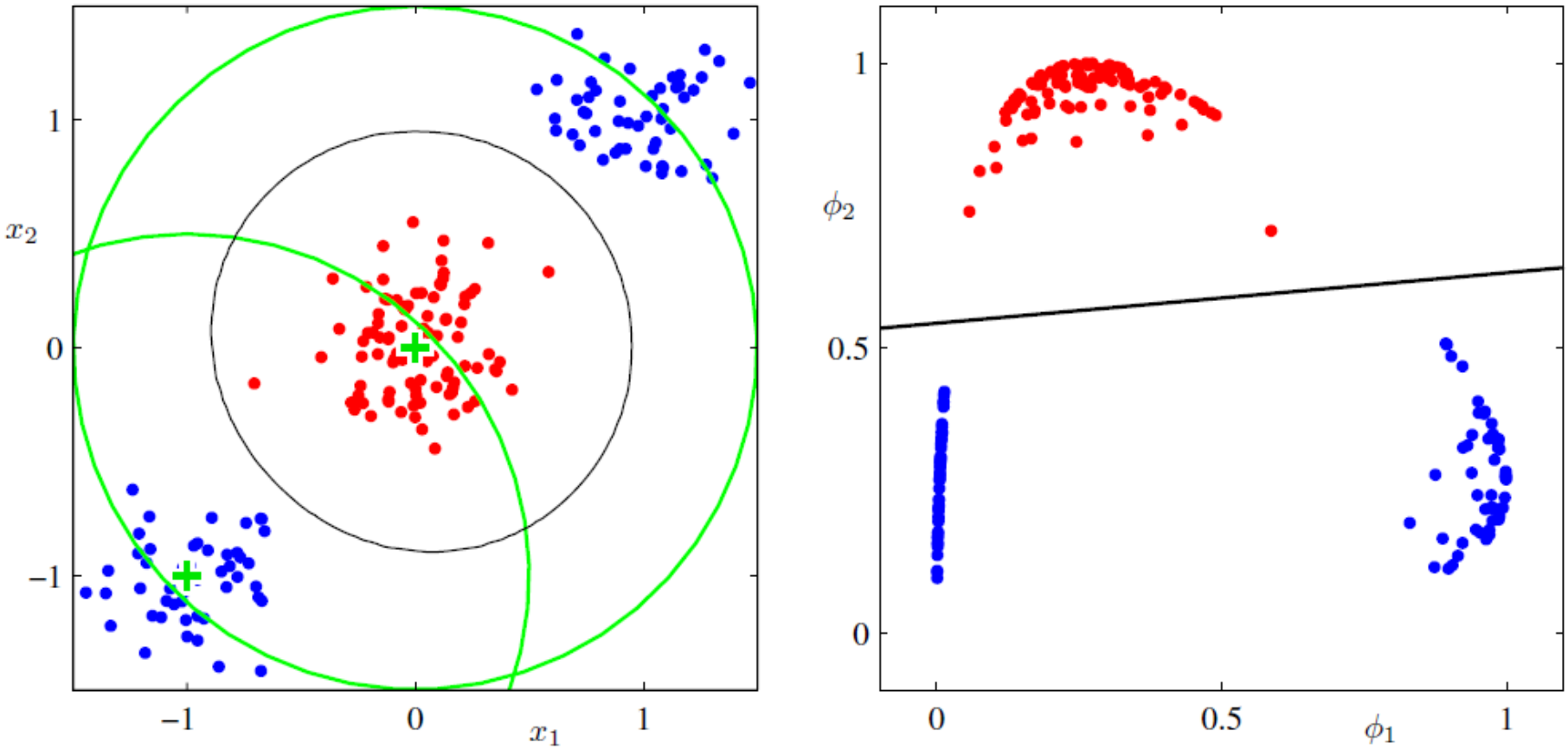
- Gradiente:

$$\nabla J(\mathbf{w}) = \frac{1}{m}\mathbf{X}^T(\tilde{\mathbf{y}} - \mathbf{y}) = \frac{1}{m}\mathbf{X}^T(\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y})$$

Extensão com Funções de Base

- Assim como no caso de regressão linear, o modelo básico de regressão logística pode ser estendido com funções de base, isto é, utilizando como atributos $x'_j = \varphi_j(\mathbf{x})$, $j = 1, \ldots, n'$, funções não-lineares dos atributos originais $\mathbf{x} = (x_1, \ldots, x_n)^T$
- O treinamento é idêntico a partir dos atributos transformados $\mathbf{x}'$, entretanto a visualização a partir dos atributos originais $\mathbf{x}$ será diferente
  - Em particular, permite uma separação não-linear entre as classes

Exemplo



Notação (Bishop): Atributos originais: x_1, x_2; Atributos transformados: ϕ_1, ϕ_2

$\phi_j(\mathbf{x}) = \exp(-\gamma\|\mathbf{x} - \mathbf{c}_j\|^2)$, $\mathbf{c}_1 = (-1, -1)$, $\mathbf{c}_2 = (0, 0)$

Regularização

- Com o aumento no número de atributos, aumenta também a tendência a overfitting no conjunto de treinamento, tornando-se importante usar regularização para garantir uma boa generalização
- Regularização ℓ_2: $\Omega(\mathbf{w}) = \frac{1}{2m}\sum_{j=1}^{n} w_j^2 = \frac{1}{2m}\mathbf{w}^T\mathbf{L}\mathbf{w}$
- Função custo:

$$\begin{aligned} J(\mathbf{w}) &= J_{\text{train}}(\mathbf{w}) + \lambda\Omega(\mathbf{w}) \\ &= \frac{1}{m}(-\mathbf{y}^T \log \tilde{\mathbf{y}} - (1-\mathbf{y})^T \log(1-\tilde{\mathbf{y}})) + \lambda\frac{1}{2m}\mathbf{w}^T\mathbf{L}\mathbf{w} \end{aligned}$$

- Gradiente:

$$\nabla J(\mathbf{w}) = \frac{1}{m}\mathbf{X}^T(\tilde{\mathbf{y}} - \mathbf{y}) + \lambda\frac{1}{m}\mathbf{L}\mathbf{w}$$

- λ é um hiperparâmetro a ser determinado na etapa de validação

Classificação Multi-Classe

Classificação Multi-Classe

- A regressão logística é, na verdade, um método de encontrar um discriminante linear $z = \mathbf{w}^T \mathbf{x}$ para um classificador binário
- Na classificação multi-classe linear, a regra de decisão é dada por

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \ \mathbf{w}_k^T \mathbf{x}$$

- Uma forma de utilizar regressão logística para encontrar os vetores $\mathbf{w}_k$ é treinando-se, independentemente, para cada classe k, um classificador que prevê o rótulo $y_k = 1[y = k]$
- Este tipo de classificador é conhecido como "um contra todos" (*one-vs-all*, *one-vs-rest*)

Classificação Multi-Classe

- Uma forma de encontrar os rótulos y_k para cada amostra é realizando a chamada binarização do rótulo y
  - Também chamada de codificação 1-de-K ou *One-Hot Encoding*

y	$\mathbf{y} = (y_1, \ldots, y_K)$
1	$(1, 0, 0, 0, \ldots, 0)$
2	$(0, 1, 0, 0, \ldots, 0)$
3	$(0, 0, 1, 0, \ldots, 0)$
$\vdots$	$\vdots$
K	$(0, 0, 0, 0, \ldots, 1)$

Regressão Softmax

- Também conhecida como regressão logística multinomial
- Consiste em ajustar um modelo com K saídas

$$\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_K), \qquad \tilde{y}_k = \text{softargmax}(\mathbf{w}_1^T\mathbf{x}, \ldots, \mathbf{w}_K^T\mathbf{x})_k$$

onde

$$\text{softargmax}(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

tendo como rótulo o vetor $\mathbf{y} = (y_1, \ldots, y_k)$, com $y_k = 1[y = k]$, o qual corresponde à binarização de y

- Tipicamente é usada como função perda a entropia cruzada categórica

$$L(\mathbf{y}, \tilde{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \tilde{y}_k$$

Terminologia

- A função $\mathrm{softargmax}$ realiza uma aproximação suave da função argmax (com *one-hot encoding*): se $z_k \gg z_j$, para todo $j \neq k$, então

$$\mathrm{softargmax}(\mathbf{z}) \approx (0, \ldots, 0, 1, 0, \ldots, 0)$$

  onde o 1 aparece na k-ésima posição.

- No entanto, na literatura esta função é normalmente denominada simplesmente função softmax

- A função que faz uma aproximação suave da função $\max$ é na verdade a função LogSumExp

$$\mathrm{LSE}(x_1, \ldots, x_n) = \log(e^{x_1} + \cdots e^{x_n}) \approx \max\{x_1, \ldots, x_n\}$$

- Confusamente, o livro do Watt utiliza a terminologia *softmax* para denotar a função LogSumExp, o que (embora faça sentido) é incomum

Avaliação de Classificadores Binários

Avaliação do modelo

- A função custo usada no treinamento (mesmo sem regularização) não necessariamente é representativa do verdadeiro custo do modelo em uma aplicação real
  - Ex: podemos estar interessados na acurácia = $1 -$ taxa de erro
- Para um classificador binário, uma avaliação genérica do modelo (sem se comprometer com uma métrica específica) pode ser feita a partir de duas grandezas:

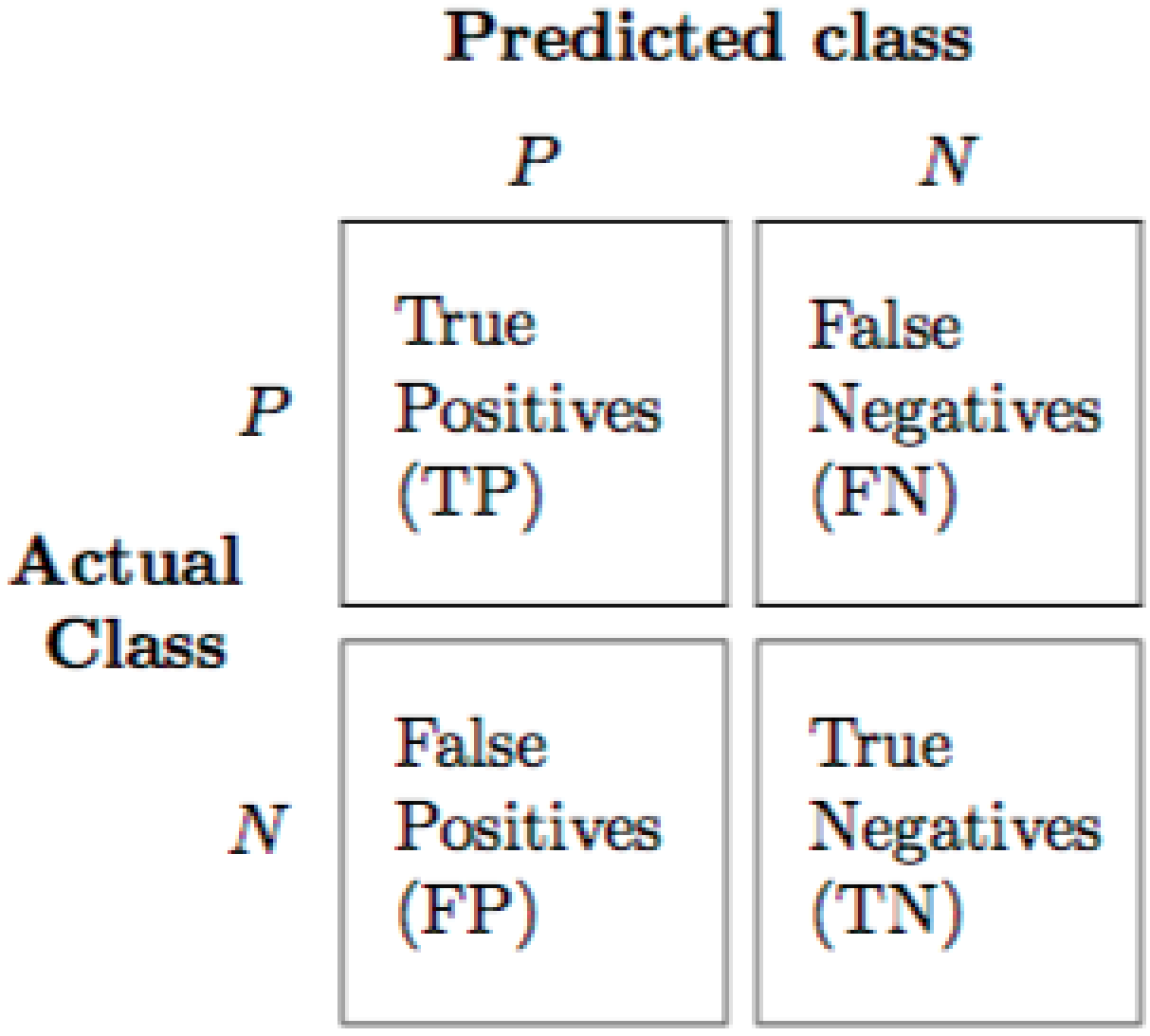
$$TPR = \frac{TP}{TP + FN} \quad \text{(true positive rate)}$$

$$FPR = \frac{FP}{FP + TN} \quad \text{(false positive rate)}$$

- Um tradeoff entre as duas grandezas (conhecido como curva ROC) pode ser obtido variando-se o limiar de decisão Δ, i.e.,

$$\hat{y} = 1[z > \Delta]$$

Matriz de confusão



Curva ROC (Receiver Operating Characteristic)