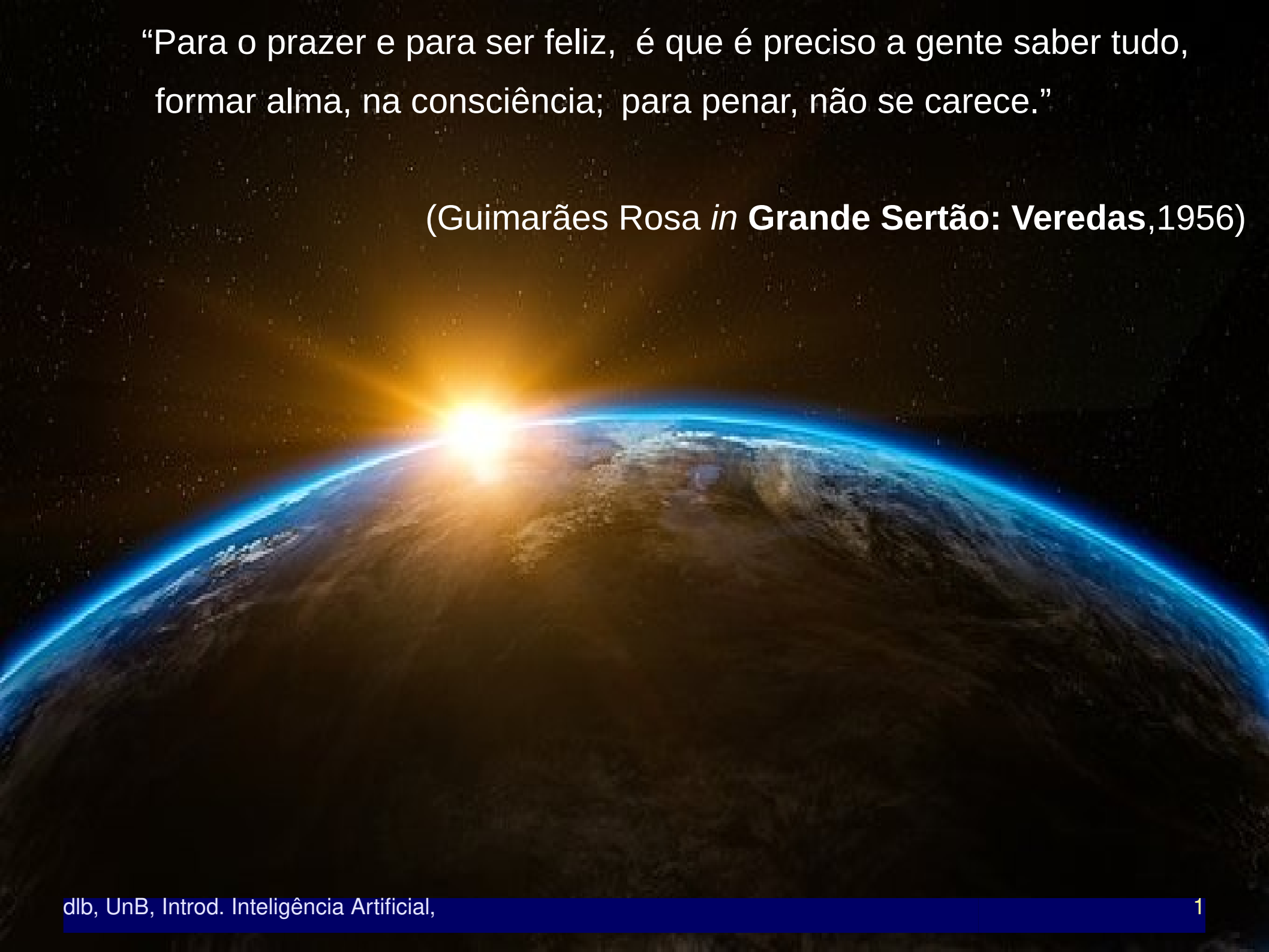


“Para o prazer e para ser feliz, é que é preciso a gente saber tudo,
formar alma, na consciência; para penar, não se carece.”

(Guimarães Rosa *in* **Grande Sertão: Veredas**, 1956)



Introdução à Inteligência Artificial

Roteiro da aula:

- ♦ **Componentes de Aprendizagem Supervisionada;**
- ♦ **Regressão por mínimos quadrados;**
 - ♦ Solução analítica fechada;
 - ♦ Solução por gradiente descendente;
- ♦ **Métricas de desempenho;**
- ♦ **Aprend. Supervisionada (com ruído);**
- ♦ **Regressão/Classificação;**
- ♦ **Regressão Logística;**
- ♦ **Classificadores LDA e QDA;**
- ♦ **Exemplos;**

Com slides adaptados de E.Eaton(UPenn)

Componentes de Aprendizagem Supervisionada

Formalização

- **Entrada:**
- **Saída:**
- **Função alvo (ideal):**
- **Dados:**
- **Hipótese:**

Componentes de Aprendizagem Supervisionada

Formalização

- **Entrada:** \mathbf{x}
- **Saída:** y
- **Função alvo (ideal):**
- **Dados:**

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$



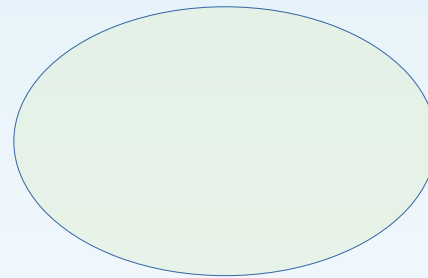
- **Hipótese:**

$$g : \mathcal{X} \rightarrow \mathcal{Y}$$

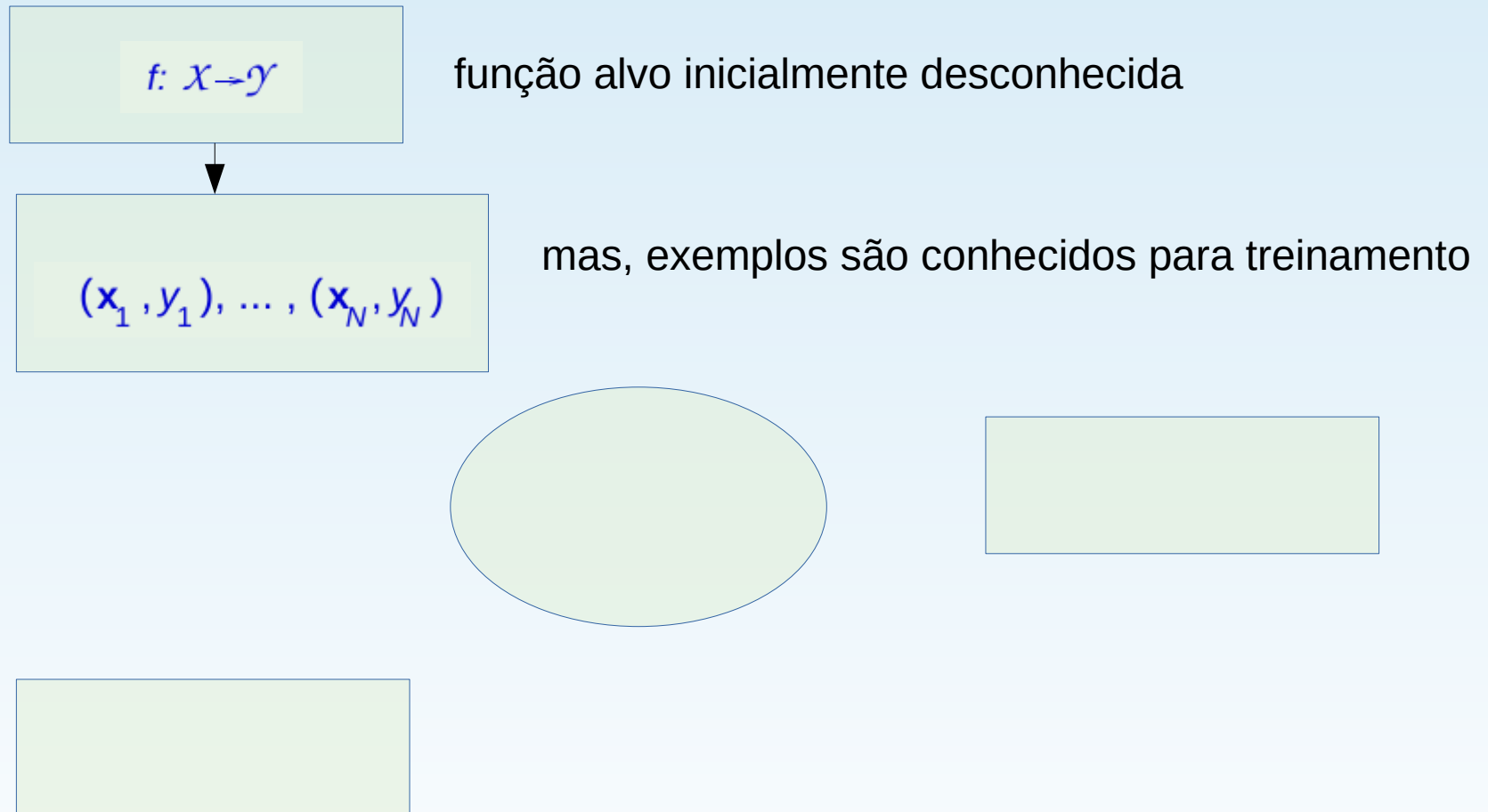
Componentes de Aprendizagem Supervisionada

$$f: \mathcal{X} \rightarrow \mathcal{Y}$$

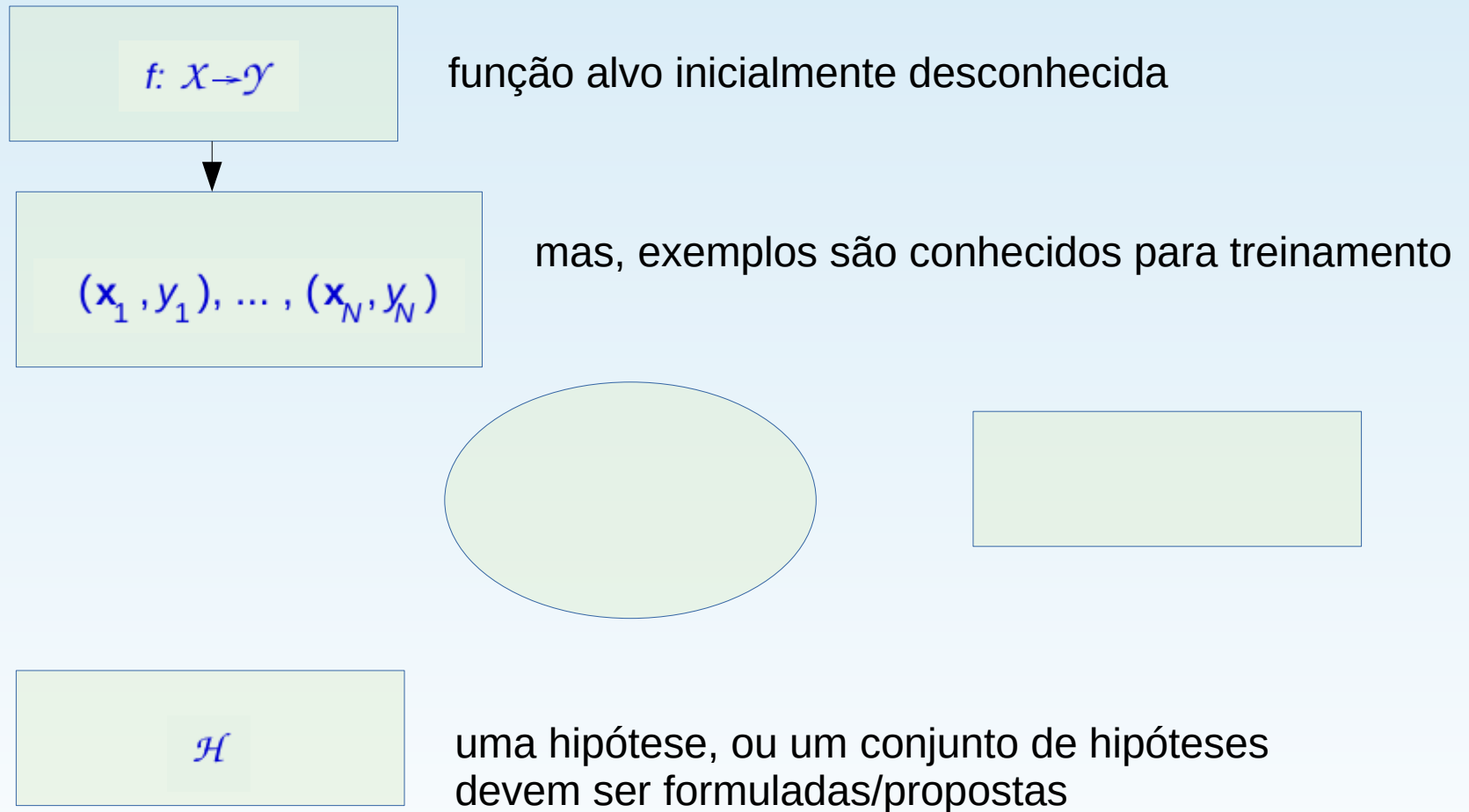
função alvo inicialmente desconhecida



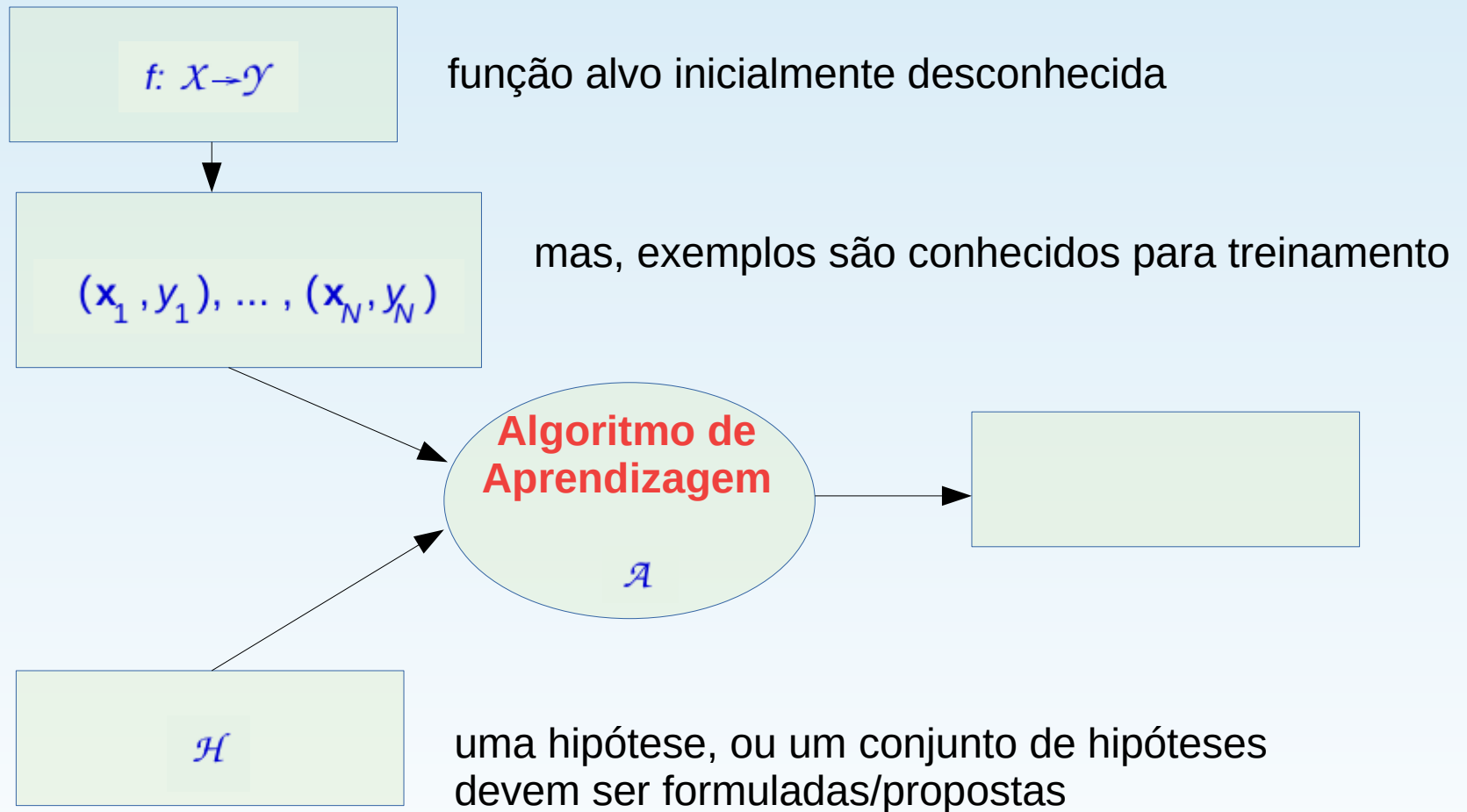
Componentes de Aprendizagem Supervisionada



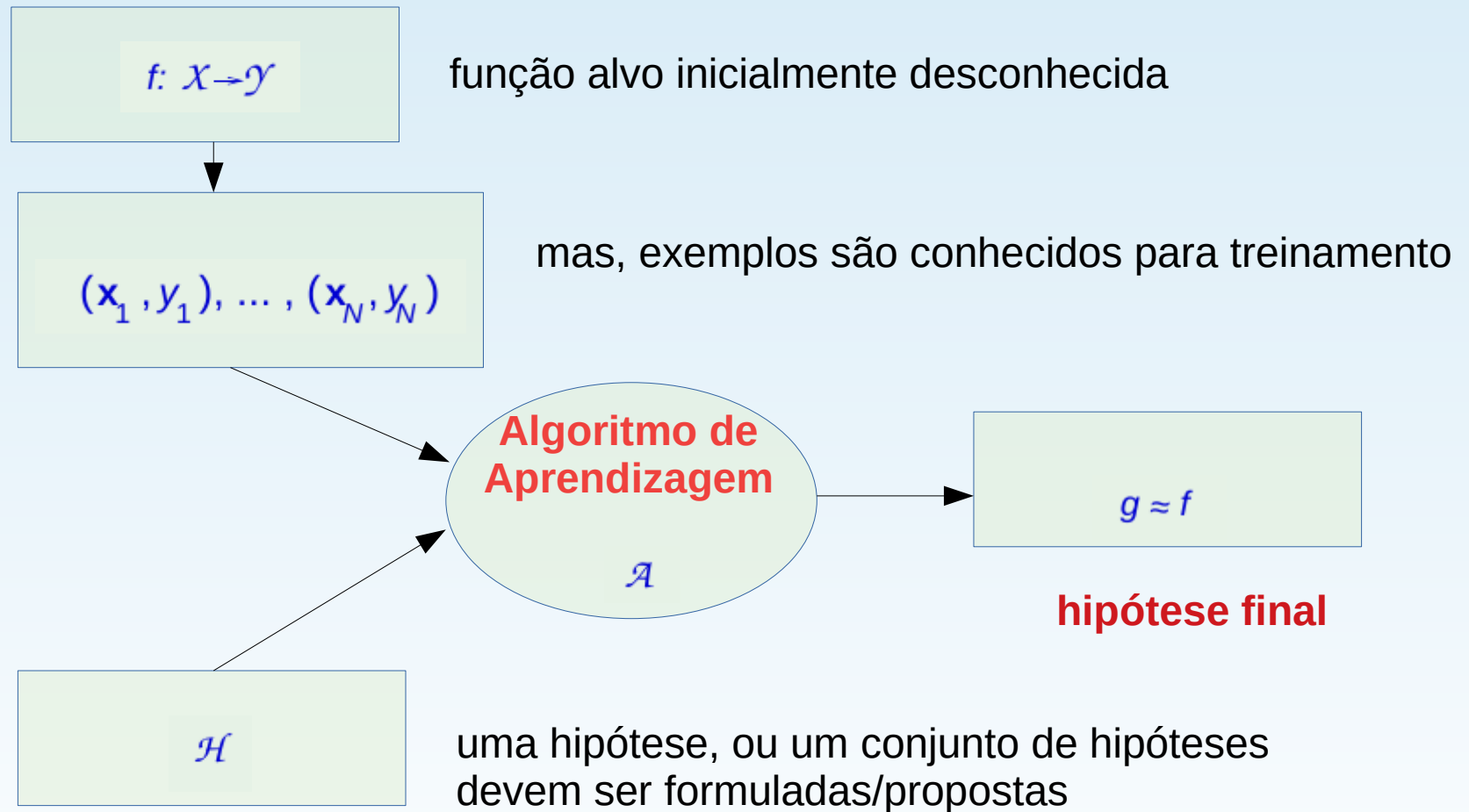
Componentes de Aprendizagem Supervisionada



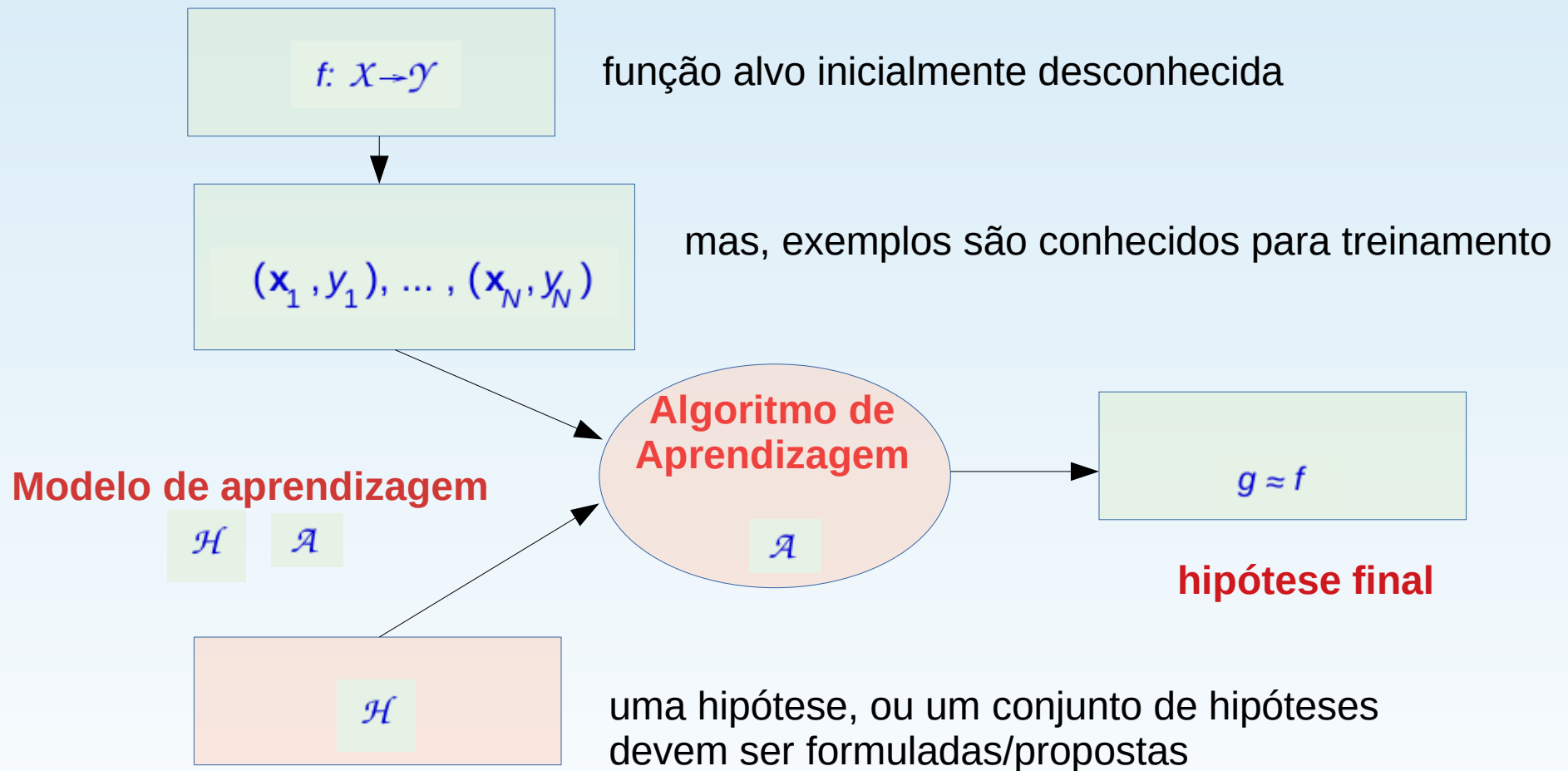
Componentes de Aprendizagem Supervisionada



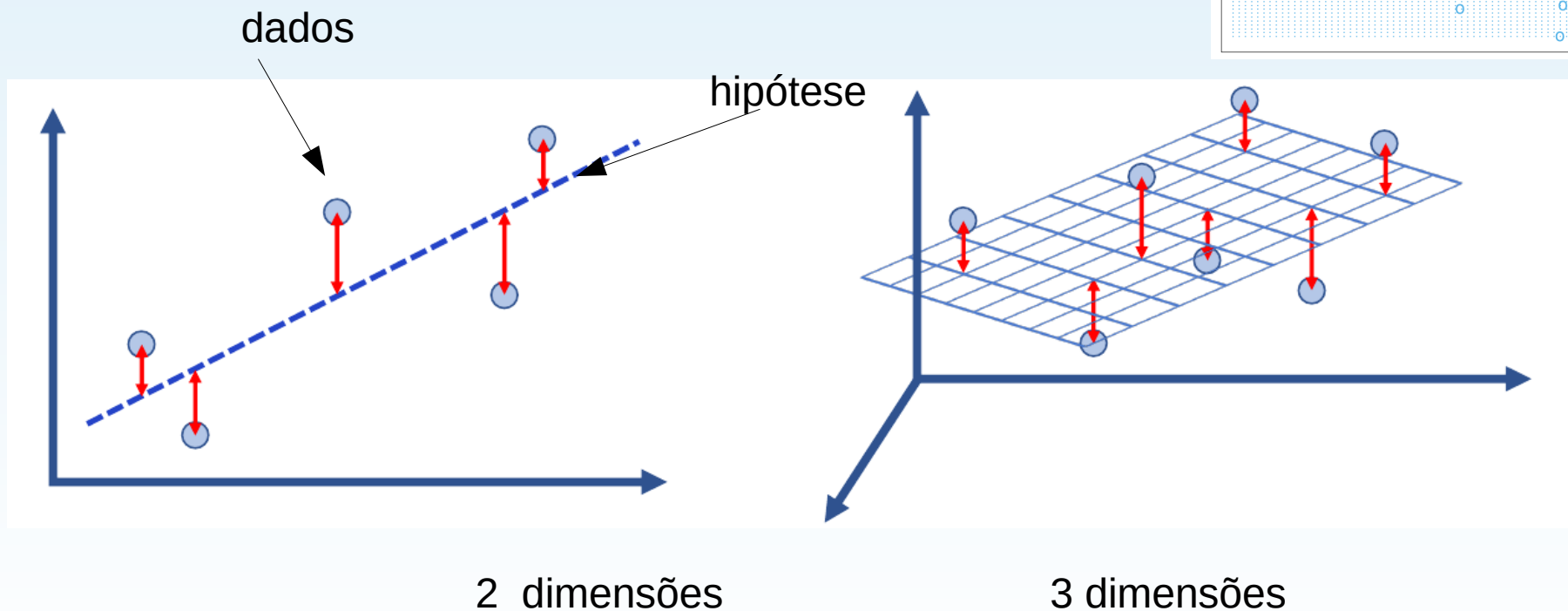
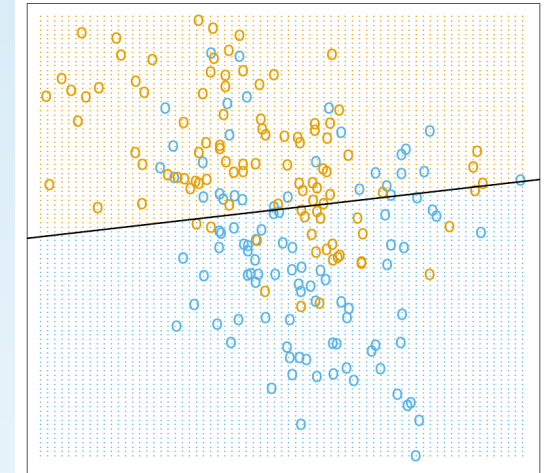
Componentes de Aprendizagem Supervisionada



Componentes de Aprendizagem Supervisionada



Regressão por mínimos quadrados



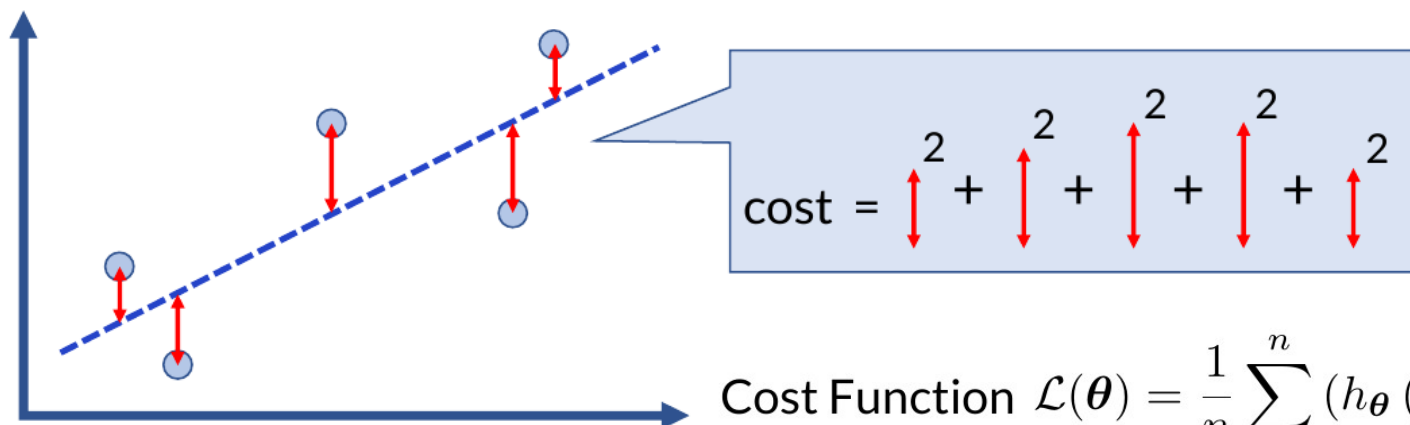
Regressão por mínimos quadrados

- Hypothesis:

$$y = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \dots + \theta_d x_{id} = \sum_{j=0}^d \theta_j x_{ij}$$

Assume $x_{i0} = 1$

- Fit model by minimizing sum of squared errors



$$\text{Cost Function } \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n (h_{\theta}(\mathbf{x}_i) - y_i)^2$$

Fit by solving $\min_{\theta} \mathcal{L}(\theta)$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

- Consider our model $h(\mathbf{x}_i) = \sum_{j=0}^d \theta_j x_{ij}$ for n instances:

- Let $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \in \mathbb{R}^{(d+1) \times 1}$ and $\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i,1} & \dots & x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,d} \end{bmatrix} \in \mathbb{R}^{n \times (d+1)}$

- Can write the model in vectorized form as $h_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{X}\boldsymbol{\theta}$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Let:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{n} \sum_{i=1}^n (h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\boldsymbol{\theta}^\top \mathbf{x}_i - y_i)^2 \\ &= \frac{1}{n} \underbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top}_{\mathbb{R}^{1 \times n}} \underbrace{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}_{\mathbb{R}^{n \times 1}} \end{aligned}$$

$\mathbb{R}^{n \times (d+1)}$
 $\mathbb{R}^{(d+1) \times 1}$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Closed Form Solution

Idea: Solve for optimal θ analytically

- Notice that the solution is when $\frac{\partial}{\partial \theta} \mathcal{L}(\theta) = 0$

• Derivation:

$$\begin{aligned}\mathcal{L}(\theta) &= \frac{1}{n} (X\theta - y)^\top (X\theta - y) \\ &\propto \theta^\top X^\top X \theta - \boxed{y^\top X \theta} - \boxed{\theta^\top X^\top y} + y^\top y \\ &\propto \theta^\top X^\top X \theta - 2\theta^\top X^\top y + y^\top y\end{aligned}$$

Note: A blue arrow points from the 1×1 label to the boxed terms $y^\top X \theta$ and $\theta^\top X^\top y$.

Take derivative and set equal to 0, then solve for θ :

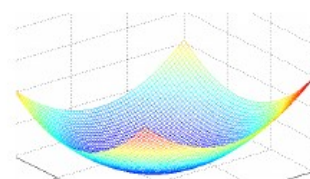
$$\frac{\partial}{\partial \theta} (\theta^\top X^\top X \theta - 2\theta^\top X^\top y + \cancel{y^\top y}) = 0$$

$$(X^\top X)\theta - X^\top y = 0$$

$$(X^\top X)\theta = X^\top y$$

Closed Form Solution:

$$\theta = (X^\top X)^{-1} X^\top y$$



Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Problems with the Closed Form Solution

- Can obtain θ by simply plugging X and y into $\theta = (X^T X)^{-1} X^T y$

$$X = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i,1} & \dots & x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,d} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

This computation is intractable for even moderate size n and d

- Computing $(X^T X)^{-1}$ is roughly $O(d^3)$
- X and/or $(X^T X)$ may be too large to fit in memory
- Numerical accuracy issues due to ill-conditioning

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Mas, pode-se buscar/otimizar para mínimos...

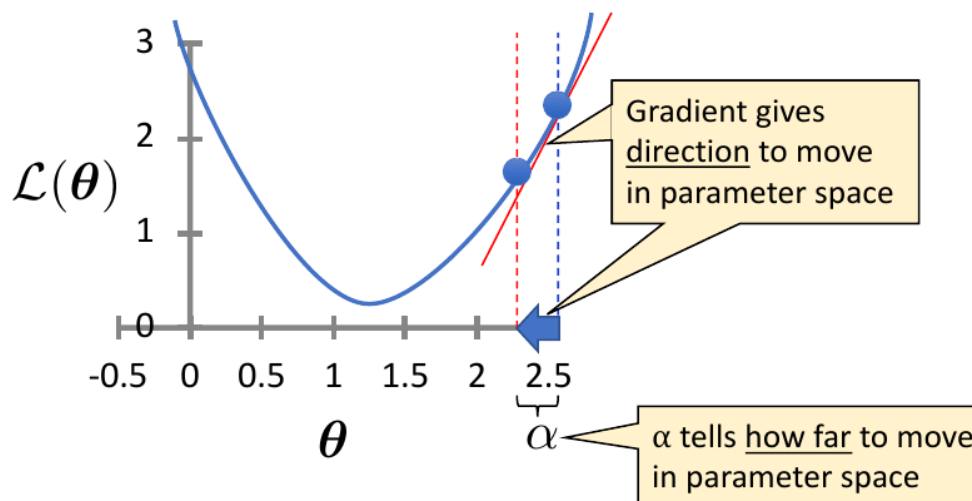
Gradient Descent

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} \mathcal{L}(\theta)$$

simultaneous update for $j = 0 \dots d$

learning rate (small)
e.g., $\alpha = 0.05$



Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Computing the Gradient for OLS

- Need to find $\frac{\partial}{\partial \theta_j} \mathcal{L}(\boldsymbol{\theta})$ for ordinary least squares:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} \mathcal{L}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \frac{1}{n} \sum_{i=1}^n (h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2 \\ &= \frac{\partial}{\partial \theta_j} \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=0}^d \theta_j x_{ij} - y_i \right)^2 \\ &= \frac{2}{n} \sum_{i=1}^n \left(\sum_{j=0}^d \theta_j x_{ij} - y_i \right) \times \frac{\partial}{\partial \theta_j} \left(\sum_{j=0}^d \theta_j x_{ij} - y_i \right) \\ &= \frac{2}{n} \sum_{i=1}^n (h_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i) x_{ij}\end{aligned}$$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Gradient Descent for Linear Regression

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{2}{n} \sum_{i=1}^n (h_{\theta}(\mathbf{x}_i) - y_i) x_{ij} \quad \text{simultaneous update for } j = 0 \dots d$$

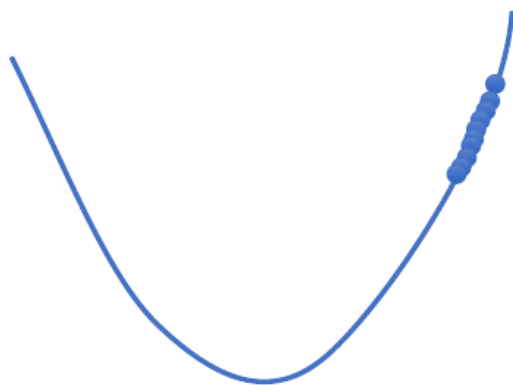
- To achieve simultaneous update
 - At the start of each GD iteration, compute $h_{\theta}(\mathbf{x}_i)$
 - Use this stored value in the update step loop
- Assume convergence when $\|\theta_{new} - \theta_{old}\|_2 < \epsilon$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

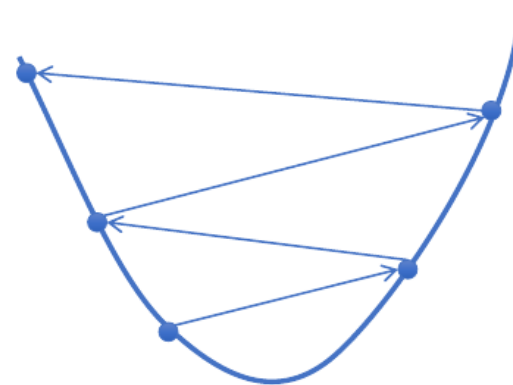
Choosing the Learning Rate (α)

slow convergence



Cause: α too small

increasing value for $\mathcal{L}(\theta)$



Cause: α too large

- To see if gradient descent is working, print out $\mathcal{L}(\theta)$ each iteration
 - The value should decrease at each iteration
 - If it doesn't, adjust α

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Gradient Descent vs Closed Form

Gradient Descent

- Requires multiple iterations
- Need to choose α
- Works well when n is large
- Can support incremental learning

Closed Form Solution

- Non-iterative
- No need for α
- Slow if n and/or d is large
 - Computing $(X^T X)^{-1}$ is roughly $O(d^3)$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

Não somente para hipóteses lineares

Extending OLS to More Complex Models

We can alter the input X in many ways:

Feature encodings	Feature transformations	Derived features
<ul style="list-style-type: none">• One-hot-encoding of categorical features• Numeric encoding of ordinal features	<ul style="list-style-type: none">• Basic functions<ul style="list-style-type: none">○ log, exp, sqrt, square, etc.• Polynomial functions<ul style="list-style-type: none">○ $x_{ij} = \beta_0 + \beta_1 \cdot x_{ij} + \beta_2 \cdot x_{ij}^2$• Basis expansions	<ul style="list-style-type: none">• Combination features<ul style="list-style-type: none">○ $x_{i,new} = x_{i1} \cdot x_{i2}$• Outputs of other ML models<ul style="list-style-type: none">○ $x_{i,new} = \text{tree.predict}(x_{i1}, x_{i2})$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

- Generally,

$$h_{\theta}(\mathbf{x}) = \sum_{j=0}^d \theta_j \underbrace{\phi_j(\mathbf{x})}_{\text{basis function}}$$

- Typically, $\phi_0(\mathbf{x}) = 1$ so that θ_0 acts as a bias
- In the simplest case, we use linear basis functions: $\phi_j(\mathbf{x}) = x_j$

- Polynomial basis functions: $\phi_j(x) = x^j$

- Gaussian basis functions:

$$\phi_j(x) = \exp \left\{ -\frac{(x - \mu_j)^2}{2s^2} \right\}$$

Image/slide credit: E.Eaton

Regressão por mínimos quadrados

- Sigmoidal basis functions:

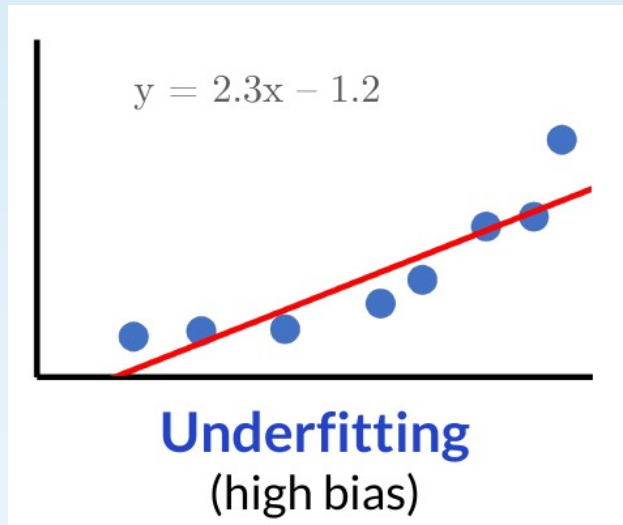
$$\phi_j(x) = \sigma \left(\frac{x - \mu_j}{s} \right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

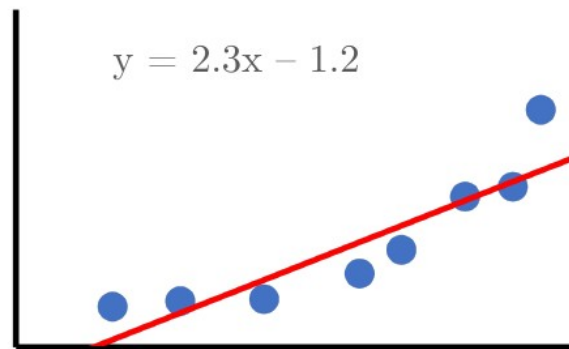
Image/slide credit: E.Eaton

Qualidade do ajuste

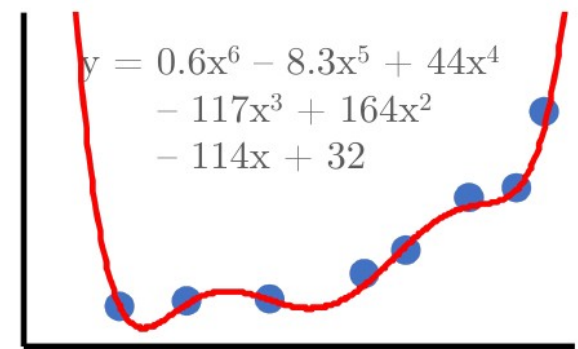


Image/slide credit: E.Eaton

Qualidade do ajuste



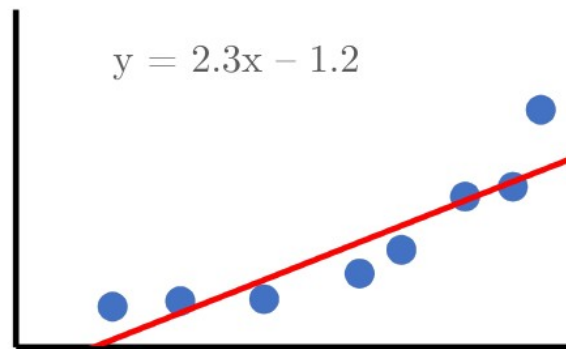
Underfitting
(high bias)



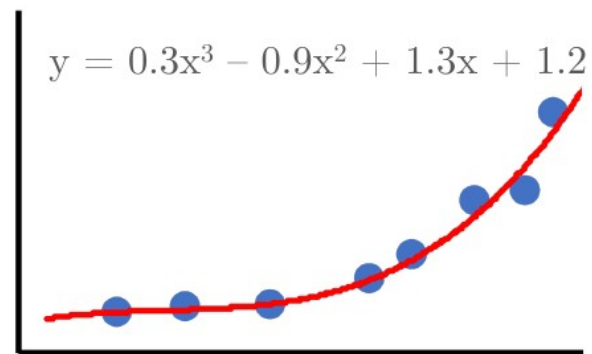
Overfitting
(high variance)

Image/slide credit: E.Eaton

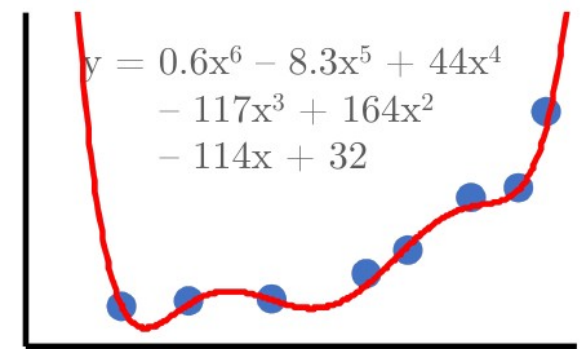
Qualidade do ajuste



Underfitting
(high bias)



Correct fit



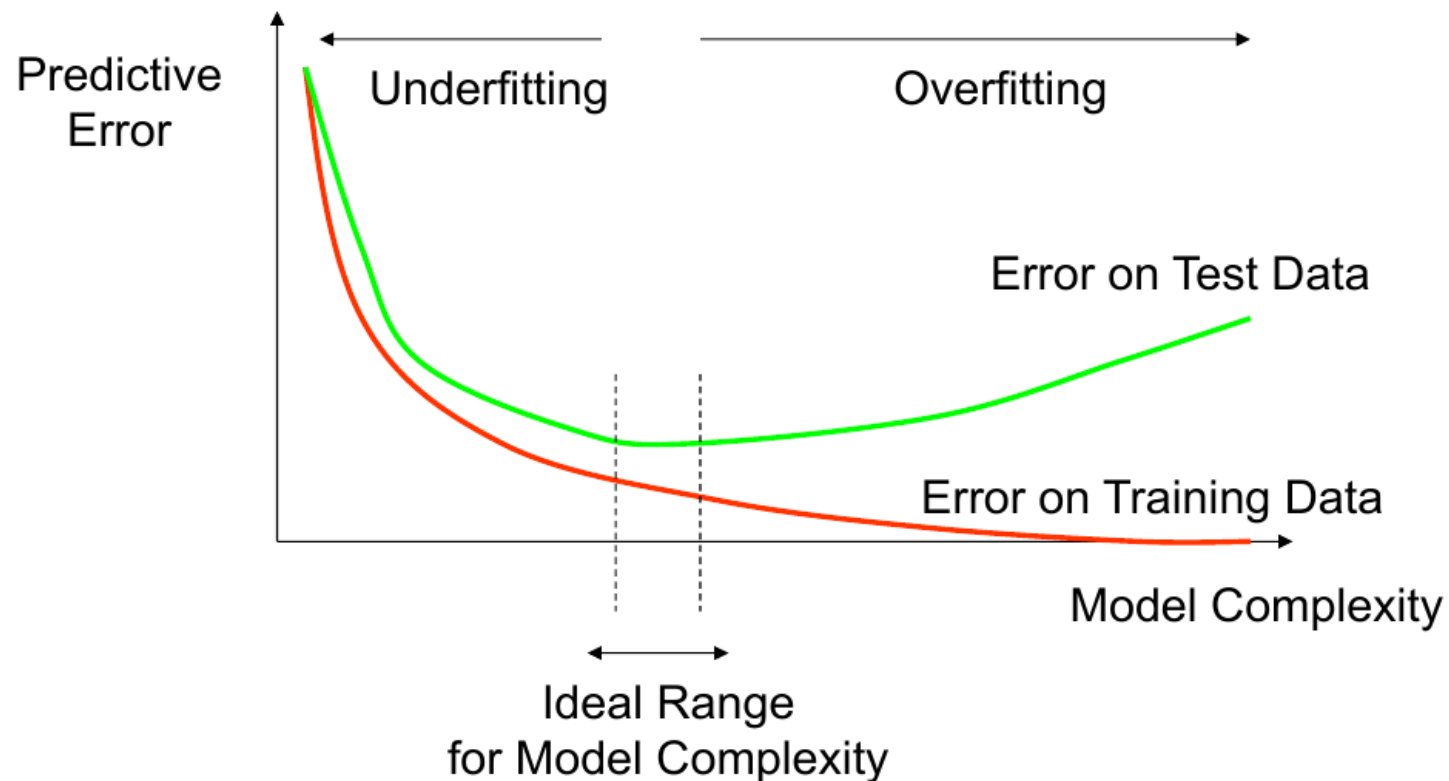
Overfitting
(high variance)

O ajuste é importante para dados futuros, predizer, generalizar

Image/slide credit: E.Eaton

Qualidade do ajuste

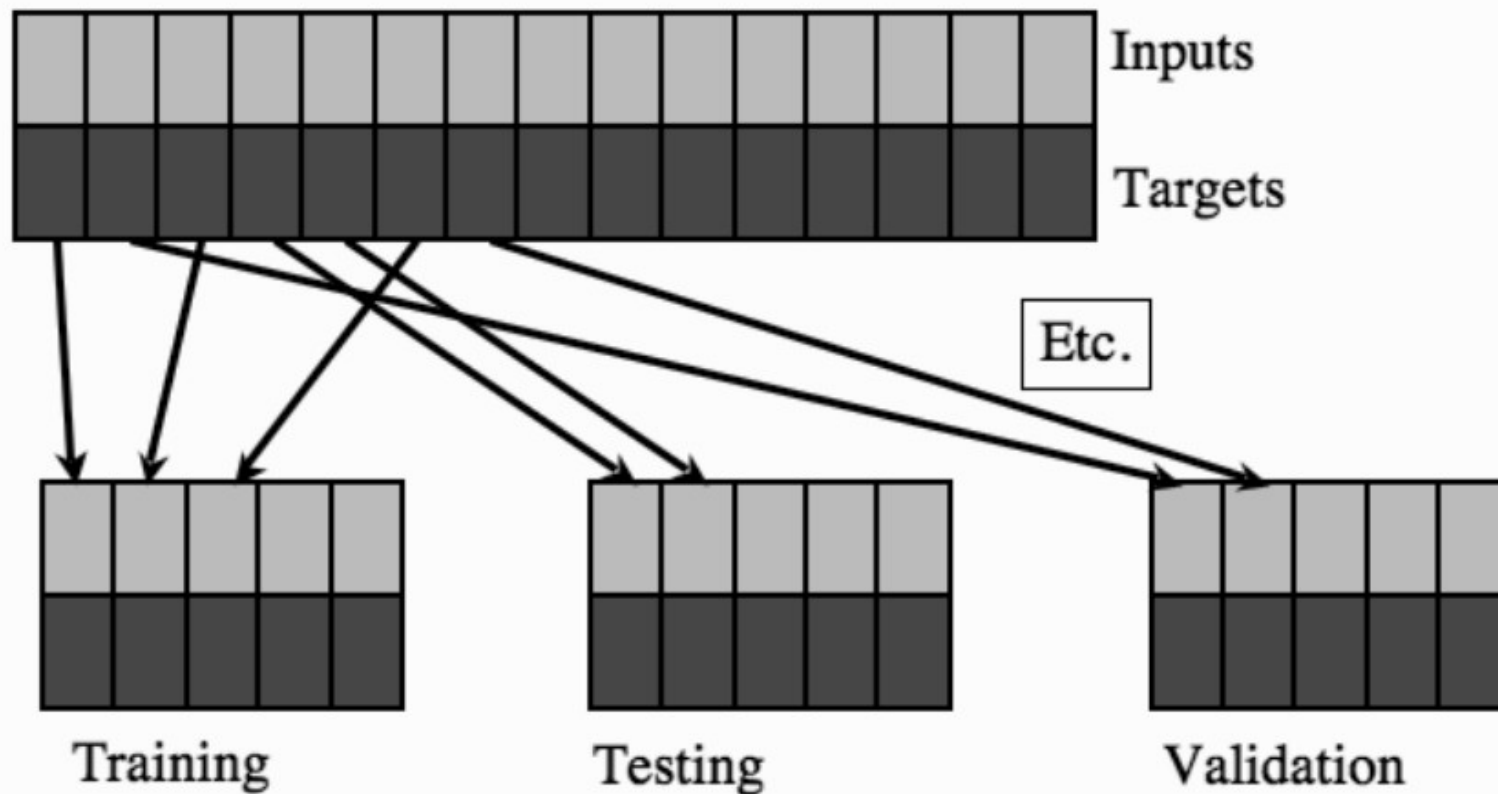
How Overfitting Affects Prediction



Image/slide credit: E.Eaton

Usando subconjuntos (randômicos) dos dados para melhorar ajuste

Dividir em 3: **Treinamento, Validação, Teste** (e.g. 50%, 20%, 30%)



(Marsland, 2015)

Usando subconjuntos (randômicos) dos dados para melhorar ajuste

Ou em validação cruzada

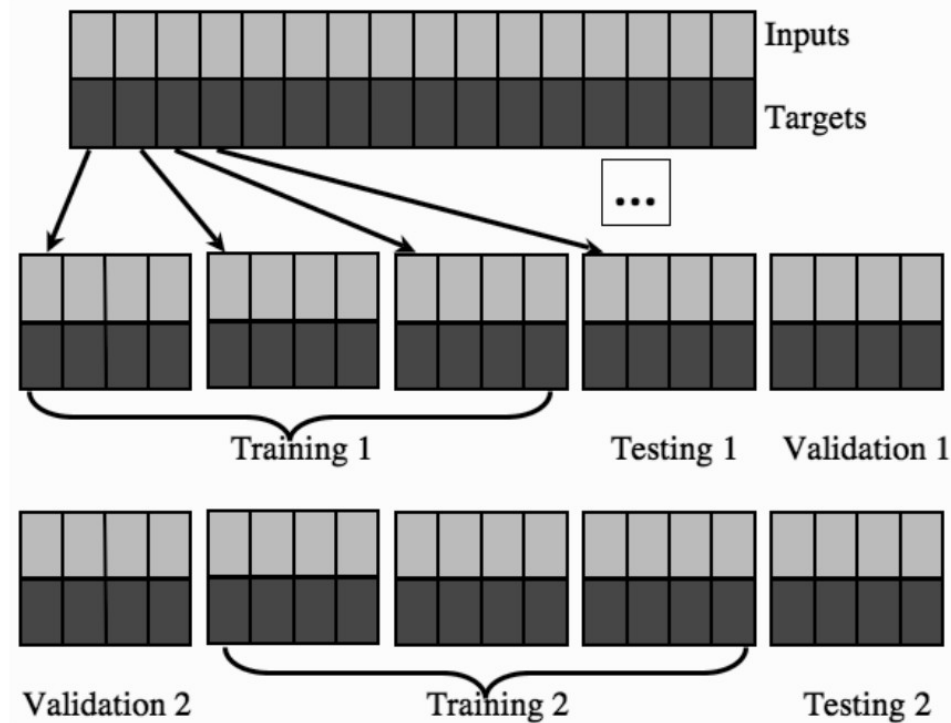


FIGURE 2.7 Leave-some-out, multi-fold cross-validation gets around the problem of data shortage by training many models. It works by splitting the data into sets, training a model on most sets and holding one out for validation (and another for testing). Different models are trained with different sets being held out.

(Marsland, 2015)

Métricas de desempenho

Matriz de confusão (nxn com resultados positivos por classes), exemplo

Outputs			
	C_1	C_2	C_3
C_1	5	1	0
C_2	1	4	1
C_3	2	0	4

Métricas de desempenho

No caso de 2 classes

True Positives	False Positives
False Negatives	True Negatives

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})}$$

Métricas de desempenho

No caso de 2 classes

True Positives	False Positives
False Negatives	True Negatives

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#FN}$$

$$\text{Specificity} = \frac{\#TN}{\#TN + \#FP}$$

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

$$F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Métricas de desempenho

Curva ROC (*Receiver Operating Characteristic*)

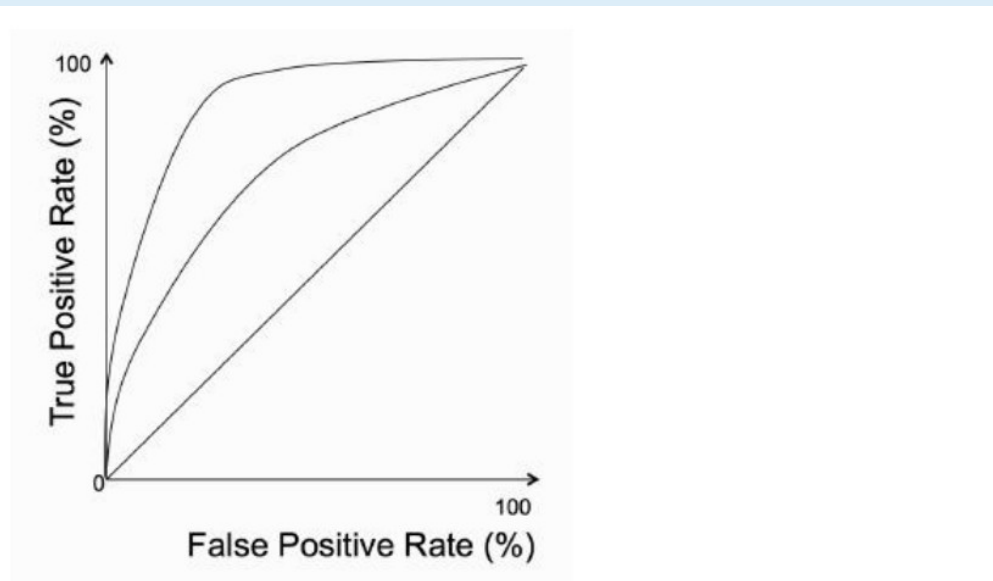


FIGURE 2.8 An example of an ROC curve. The diagonal line represents exactly chance, so anything above the line is better than chance, and the further from the line, the better. Of the two curves shown, the one that is further away from the diagonal line would represent a more accurate method.

(Marsland, 2015)

Métricas de desempenho

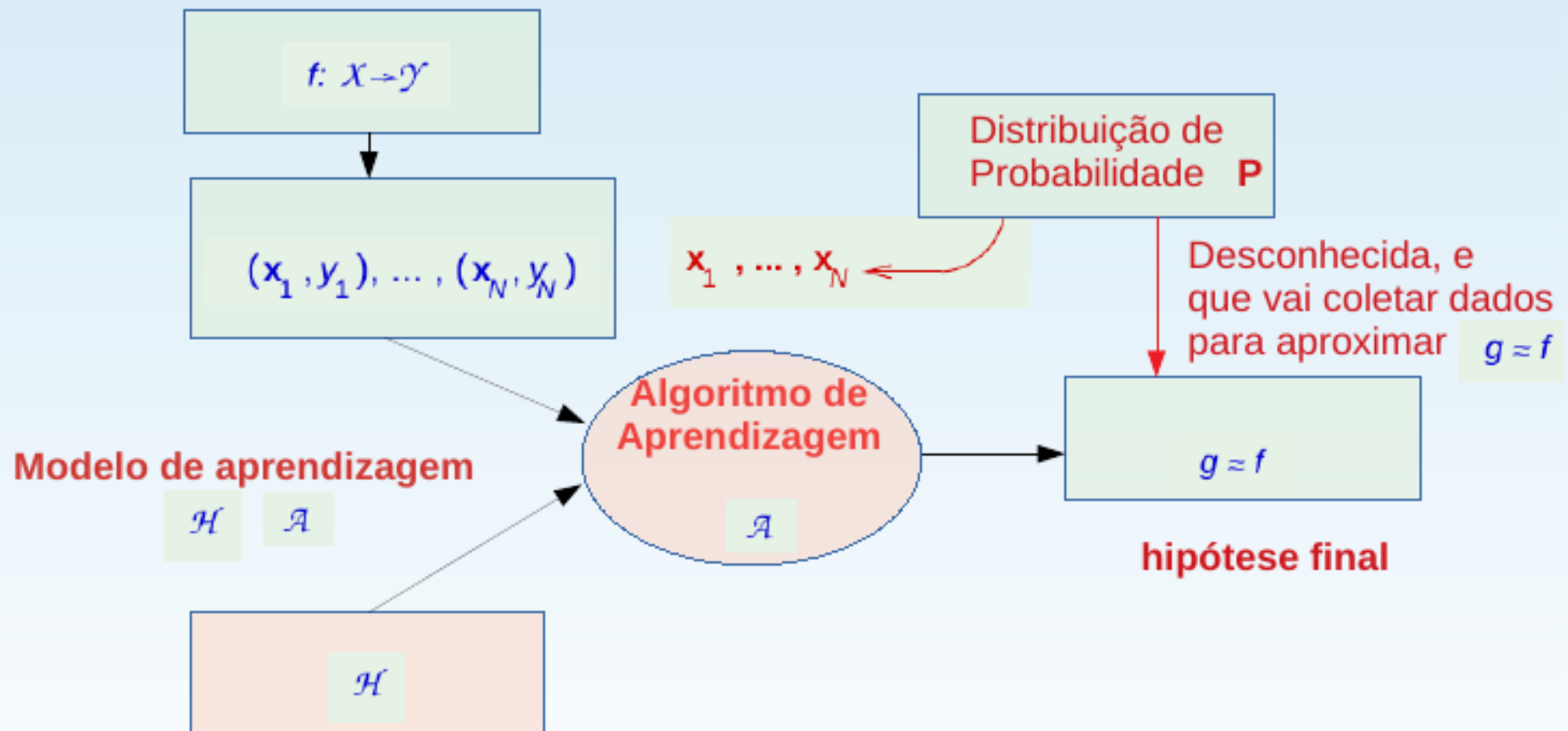
No caso de dados desbalanceados por classe, ao invés de acurácia

$$MCC = \frac{\#TP \times \#TN - \#FP \times \#FN}{\sqrt{(\#TP + \#FP)(\#TP + \#FN)(\#TN + \#FP)(\#TN + \#FN)}}$$

Matthew's Correlation Coefficient.

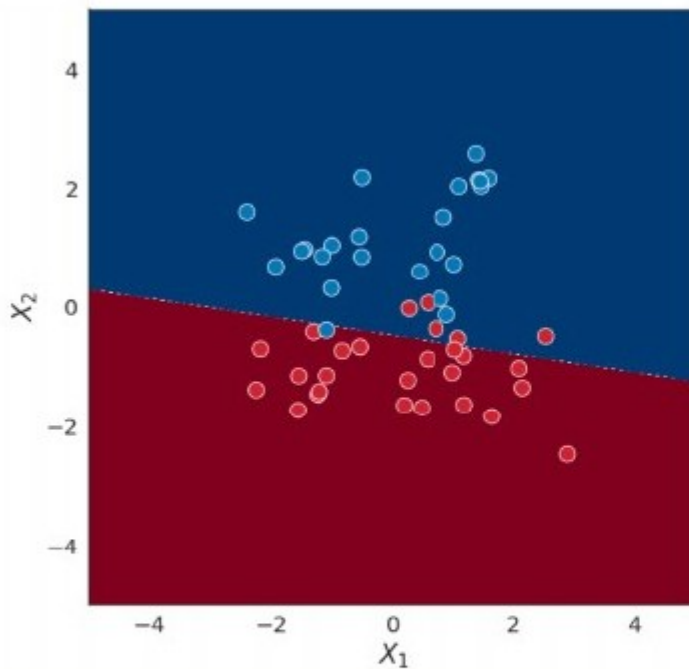
(Marsland, 2015)

Conceituação Ap. Sup. com probabilidade

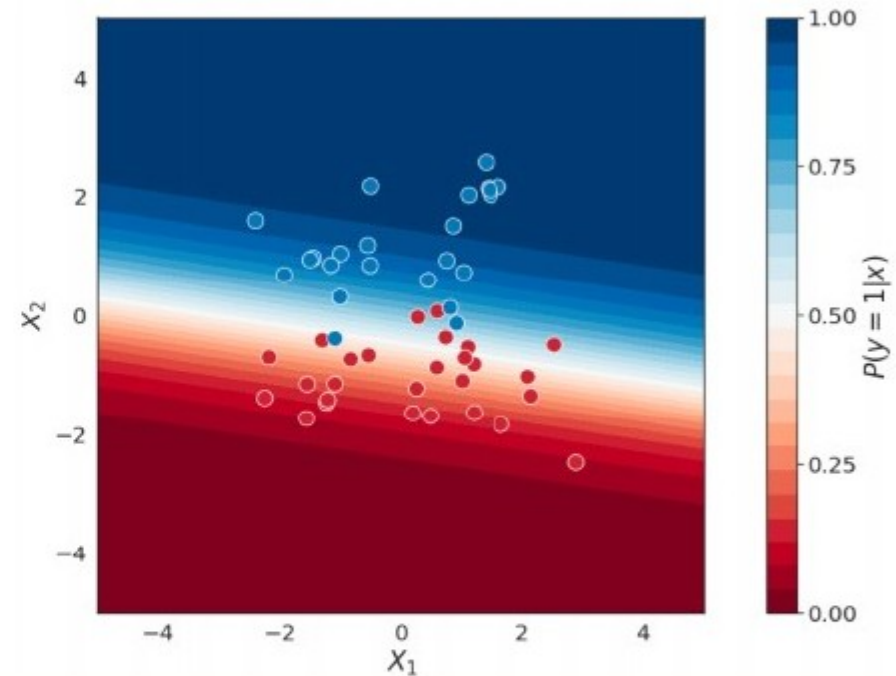


Classificação (tipos de predição)

Class prediction

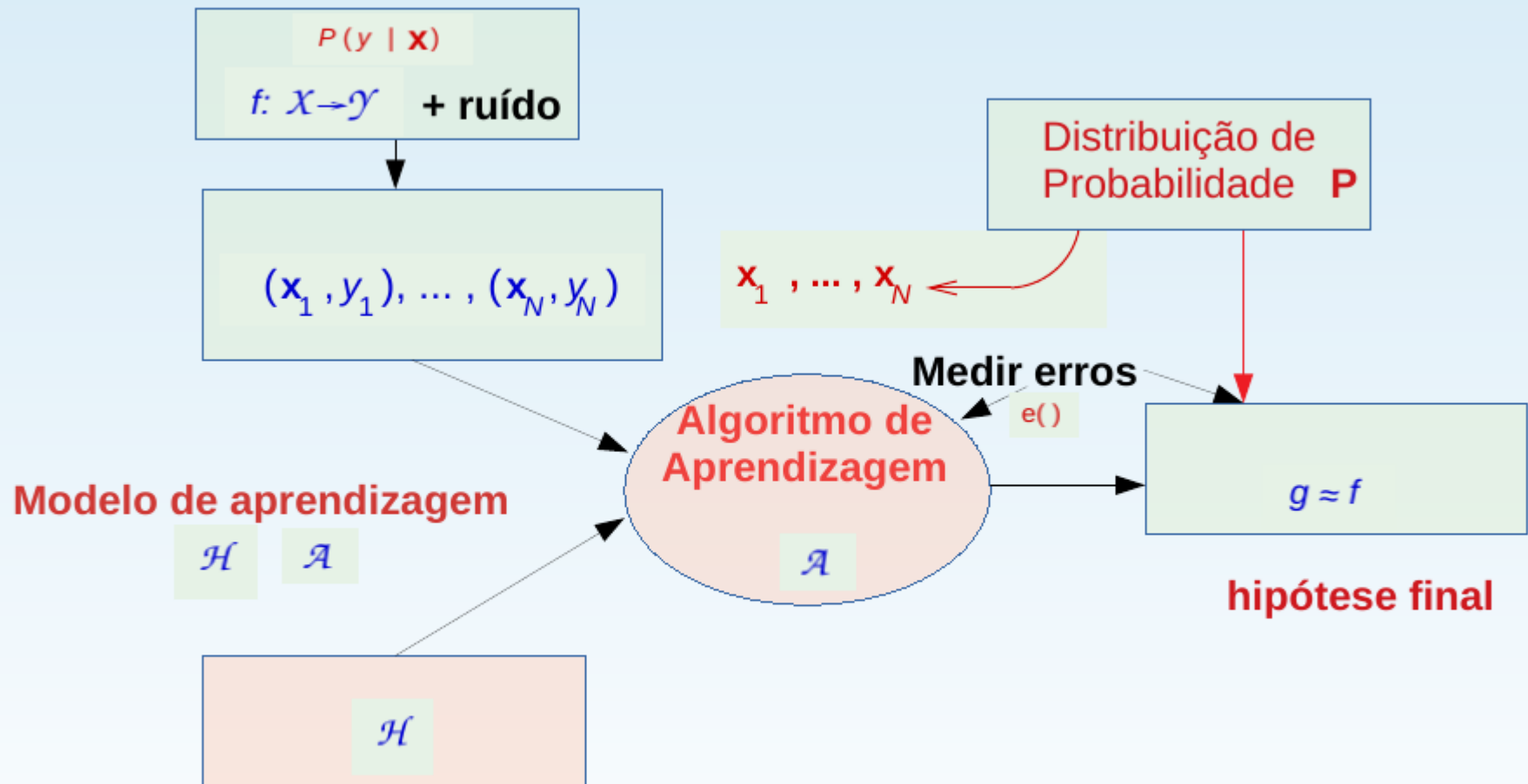


Probability prediction



Credit image: E.Eaton

Aprend. Supervisionada (com ruído)



M hipóteses e N grande...

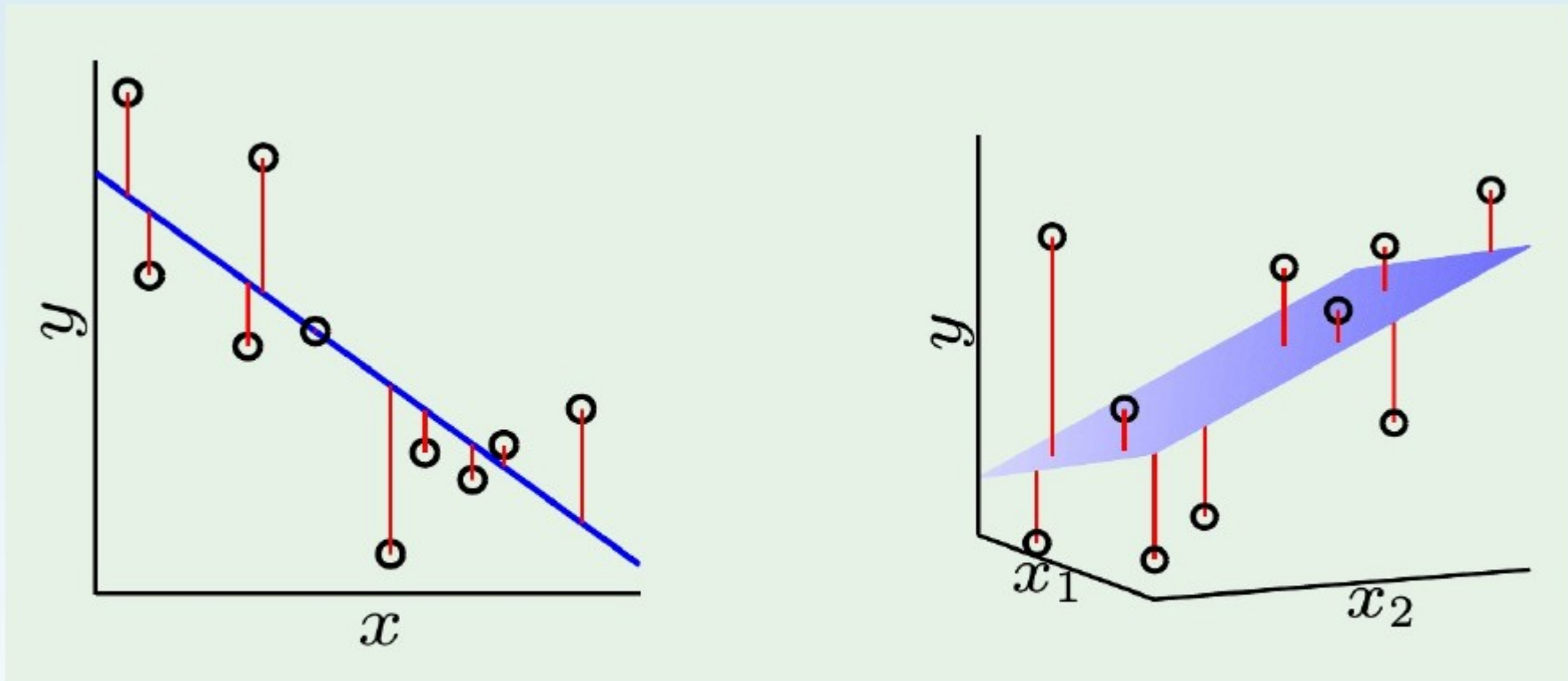
The final verdict

$$\begin{aligned}\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon] &\leq \sum_{m=1}^M \mathbb{P}[|E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon] \\ &\leq \sum_{m=1}^M 2e^{-2\epsilon^2 N}\end{aligned}$$

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

Crédito imagem: (Abu-Mostafa, 2012)

Relembrando caso linear (hipótese)



Construir essa função de decisão entre os pontos, supondo/propondo linear

De regressão para classificação

Em regressão y é **quantitativo** (em geral número real) e contínuo, para classificação precisamos de y **qualitativo**.

Como transformar?

Linear regression for classification

Linear regression learns a real-valued function $y = f(\mathbf{x}) \in \mathbb{R}$

Binary-valued functions are also real-valued! $\pm 1 \in \mathbb{R}$

Use linear regression to get \mathbf{w} where $\mathbf{w}^T \mathbf{x}_n \approx y_n = \pm 1$

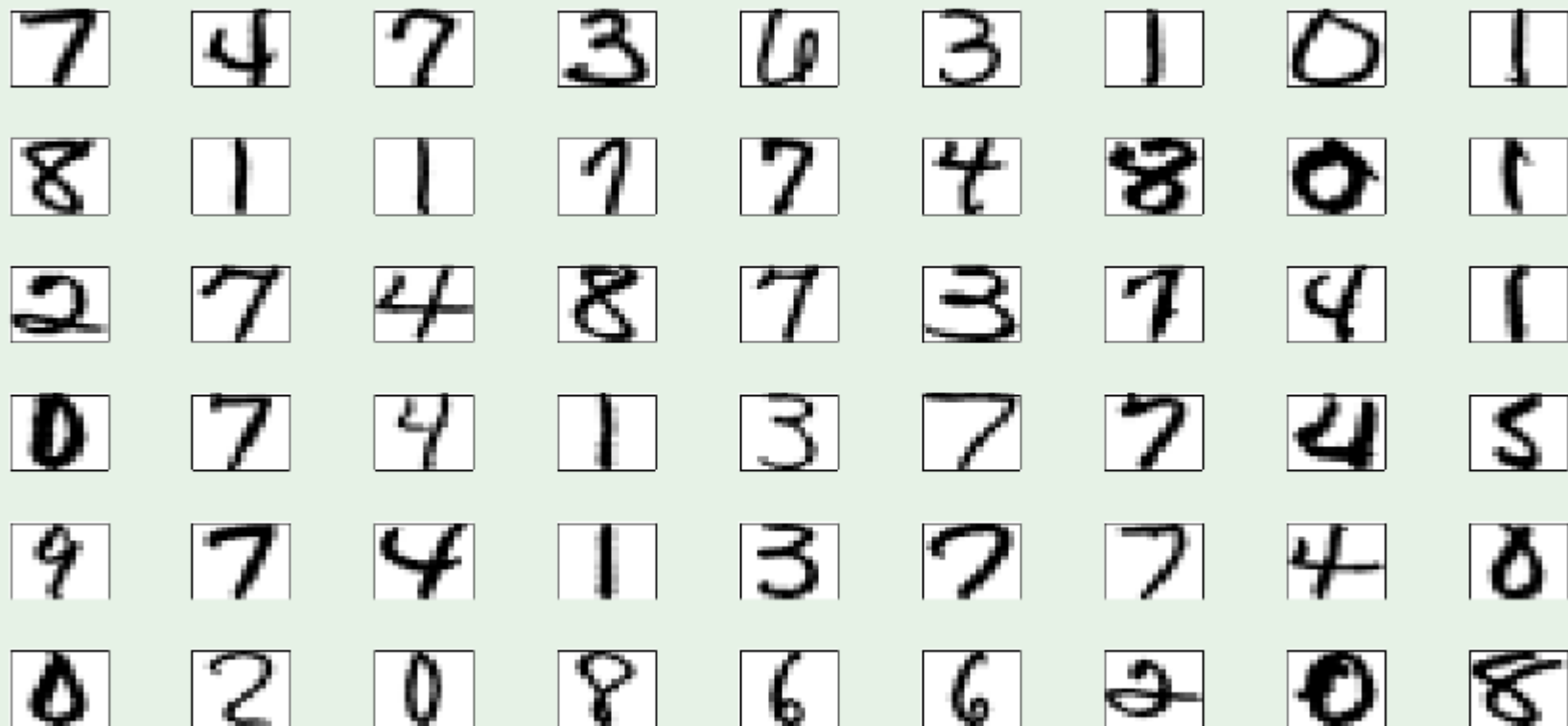
In this case, $\text{sign}(\mathbf{w}^T \mathbf{x}_n)$ is likely to agree with $y_n = \pm 1$

Good initial weights for classification

Crédito imagem: (Abu-Mostafa, 2012)

Exemplos de atributos e ruídos

A real data set



Crédito imagem: (Abu-Mostafa, 2012)

Exemplos de atributos e ruídos

Input representation

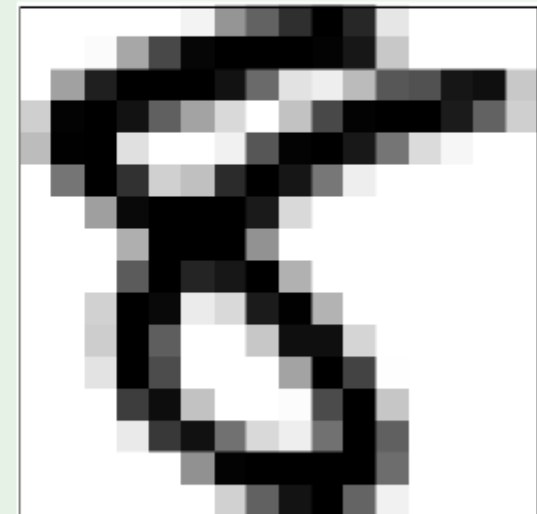
'raw' input $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{256})$

linear model: $(w_0, w_1, w_2, \dots, w_{256})$

Features: Extract useful information, e.g.,

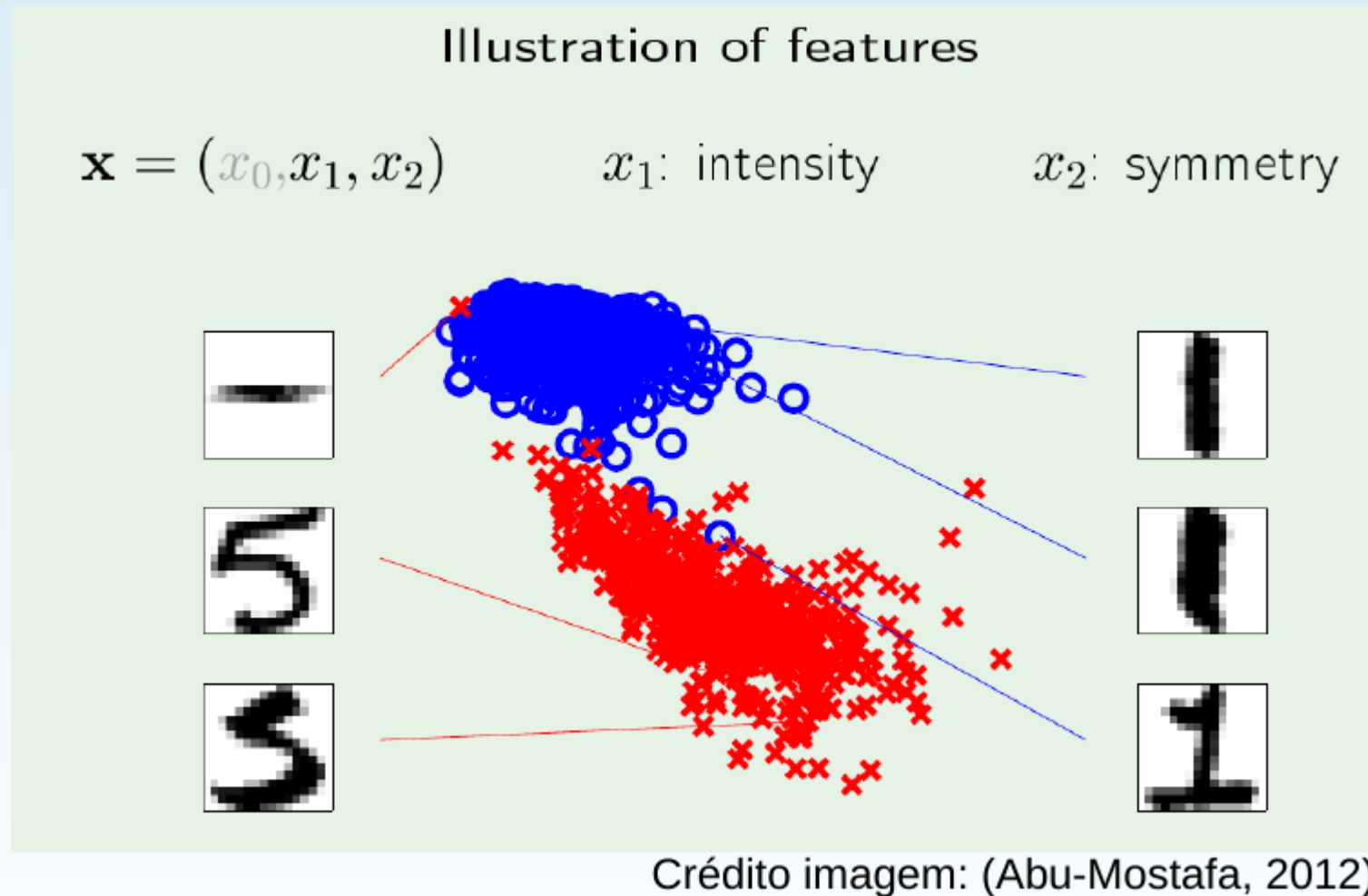
intensity and symmetry $\mathbf{x} = (x_0, x_1, x_2)$

linear model: (w_0, w_1, w_2)

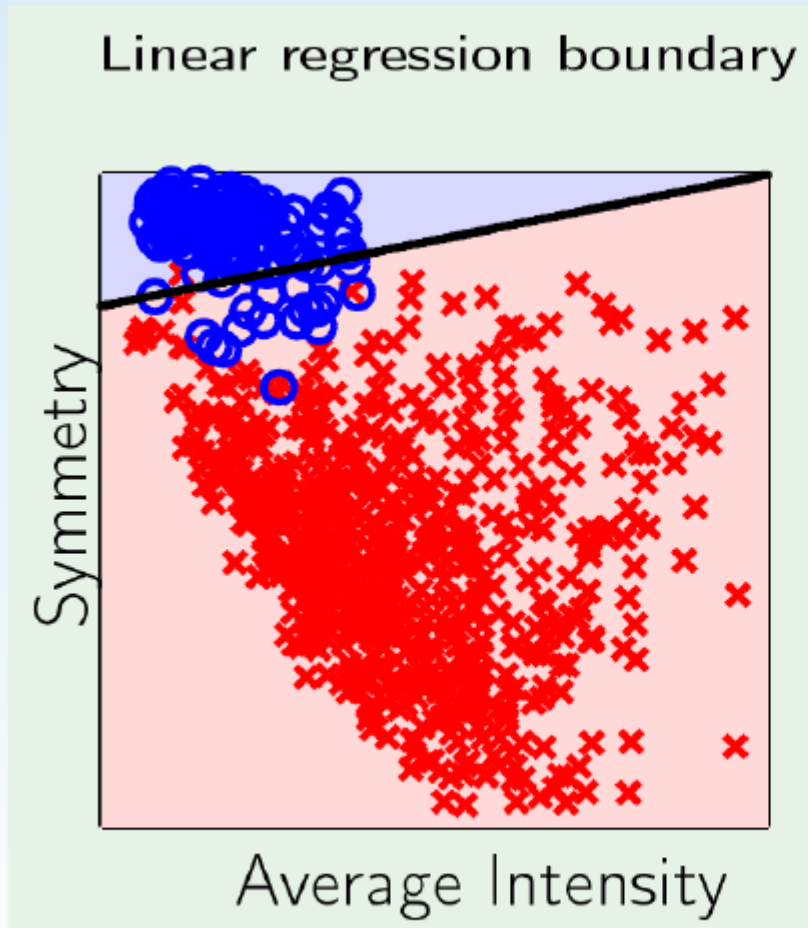


Crédito imagem: (Abu-Mostafa, 2012)

Exemplos de atributos e ruídos



Exemplos de atributos e ruídos



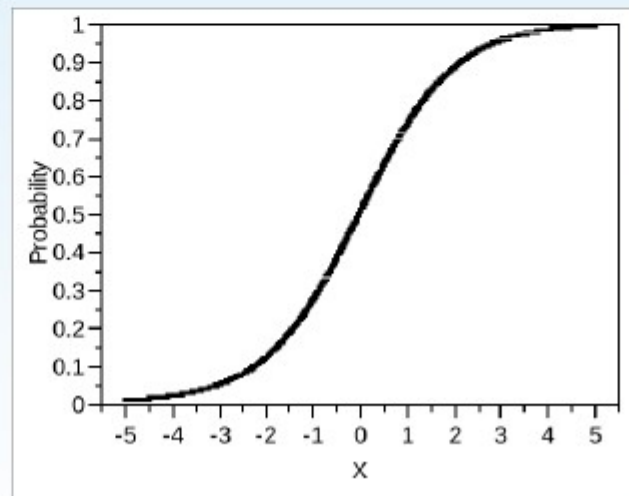
Crédito imagem: (Abu-Mostafa, 2012)

Regressão logística

- A reta de regressão $\beta_0 + \beta_1 X$ pode receber qualquer valor entre infinitamente negativo e positivo

Solução: Usar Função Logística

$$p = P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

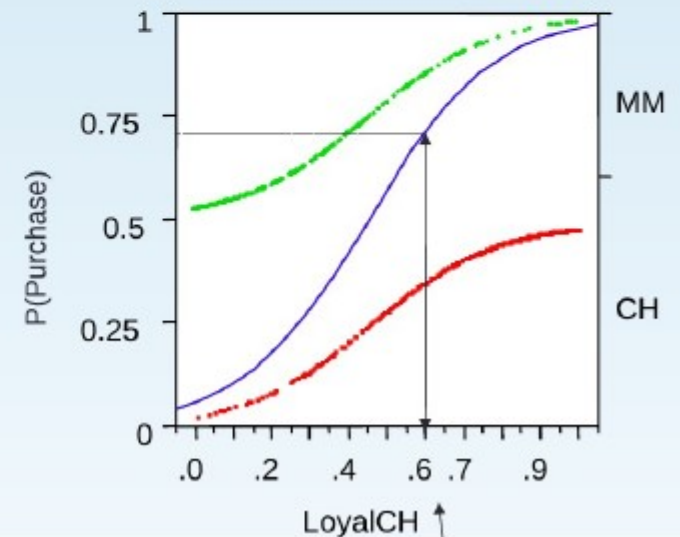


Regressão logística

Solução: Usar Função Logística

- Ao invés de tentar prever Y , vamos tentar prever $P(Y = 1)$,
- Então, podemos modelar $P(Y = 1)$ usando uma função que retorna valores entre 0 e 1.
- Podemos usar a função logística
- Regressão Logística

- Regressão logística é similar à regressão linear
- Ajustamos b_0 e b_1 para estimar β_0 e β_1 .



Regressão logística múltipla

- For 2 classes:

$$h_{\theta}(\mathbf{x}) = \frac{1}{1 + \exp(-\theta^T \mathbf{x})} = \frac{\exp(\theta^T \mathbf{x})}{\boxed{1} + \boxed{\exp(\theta^T \mathbf{x})}}$$

weight assigned to $y = 0$ weight assigned to $y = 1$

- For C classes $\{1, \dots, C\}$:

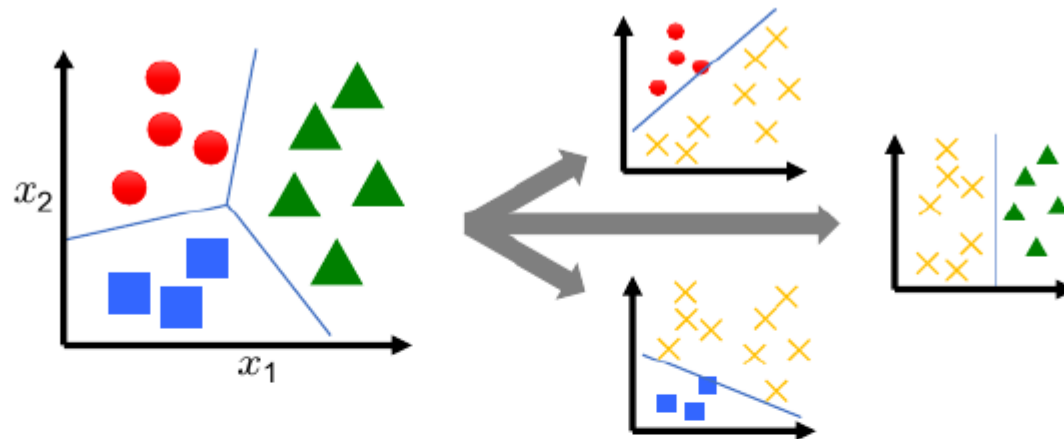
$$p(y = c \mid \mathbf{x}; \theta_1, \dots, \theta_C) = \frac{\exp(\theta_c^T \mathbf{x})}{\sum_{c=1}^C \exp(\theta_c^T \mathbf{x})}$$

- Called the **softmax** function

Credit image: E.Eaton

Regressão logística múltipla

- Split into One vs Rest:



- Train a logistic regression classifier for each class i to predict the probability that $y = i$ with

$$h_c(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_c^T \mathbf{x})}{\sum_{c=1}^C \exp(\boldsymbol{\theta}_c^T \mathbf{x})}$$

Credit image: E.Eaton

Classificador LDA (linear discriminant analysis)

Por que Linear? Por que Discriminante?

- LDA envolve a determinação de equação linear (como regressão linear) que irá predizer a qual grupo o caso pertence.

$$D = v_1X_1 + v_2X_2 + \dots + v_iX_i + a$$

- D: função discriminante
- v: coeficiente discriminante ou peso para a variável
- X: variável
- a: constante

Classificador LDA (linear discriminant analysis)

Propósito de LDA

- Escolher os v's de forma a maximizar a distância entre as médias de categorias diferentes
- Bons preditores tendem a altos v's (peso)
- Deseja-se discriminar entre categorias diferentes
- Como em uma receita. Mudando as proporções (pesos) dos ingredientes mudarão as características do produto final.

Classificador LDA (linear discriminant analysis)

Premissas de LDA

- As observações são de amostras randômicas
- Cada variável preditora é normalmente distribuída

Classificador LDA (linear discriminant analysis)

Aplicar LDA

- LDA assume que cada classe tem uma distribuição normal com uma variância comum
- Média e variância são estimadas
- Finalmente, teorema de Bayes é usado para computar p_k e a observação é atribuída à classe com a maior probabilidade entre as k probabilidades (regra de Bayes).

Classificador LDA (linear discriminant analysis)

Teorema de Bayes

$$P(i | \mathbf{x}) = \frac{P(\mathbf{x} | i) \cdot P(i)}{\sum_j P(\mathbf{x} | j) \cdot P(j)}$$

Na prática, assume-se cada grupo com distribuição normal, e que todos os grupos terão mesma matriz de covariância. O objeto k , será atribuído ao grupo i com máxima f_i .

$$f_i = \mu_i \mathbf{C}^{-1} \mathbf{x}_k^T - \frac{1}{2} \mu_i \mathbf{C}^{-1} \mu_i^T + \ln(p_i)$$

Distância de Mahalanobis

$$\mu_i \mathbf{C}^{-1} \mu_i^T$$

algoritmo LDA (linear discriminant analysis)

1. Calcular separabilidade entre classes

$$S_b = \sum_{i=1}^g N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

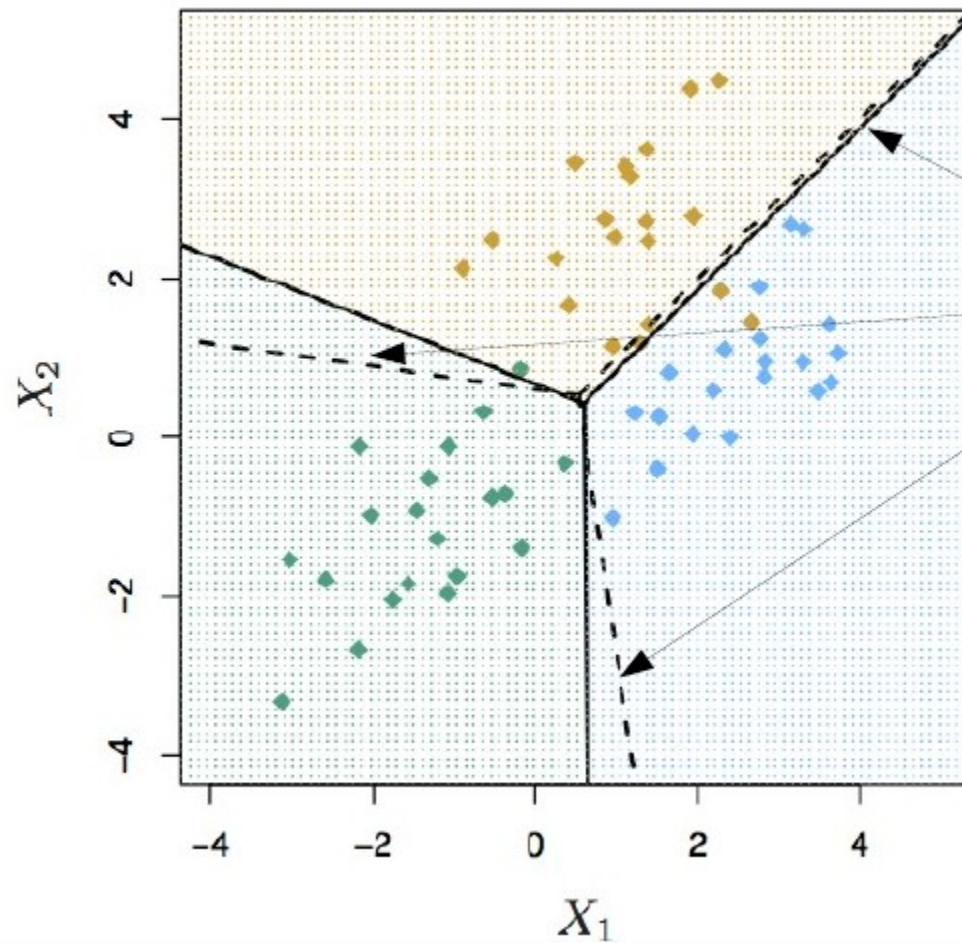
2. Calcular variância intra-classes

$$S_w = \sum_{i=1}^g (N_i - 1) S_i = \sum_{i=1}^g \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

3. Construir espaço que maximiza 1. e minimiza 2.

$$P_{lda} = \arg \max_P \frac{|P^T S_b P|}{|P^T S_w P|}$$

algoritmo LDA (linear discriminant analysis)



Exemplo

LDA

Casos não lineares?

- LDA assume que todas as classes tem a mesma variância/covariância
- Se isso não for verdadeiro, o desempenho de LDA é ruim

Casos não lineares?

- LDA assume que todas as classes tem a mesma variância/covariância
- Se isso não for verdadeiro, o desempenho de LDA é ruim

**Possibilidade seria usar
QDA (Quadratic Discriminant Analysis)**

- QDA é similar à LDA exceto que estima variâncias e covariâncias separadas para cada classe

Casos não lineares?

**Possibilidade seria usar
QDA (Quadratic Discriminant Analysis)**

- QDA é similar à LDA exceto que estima variâncias e covariâncias separadas para cada classe

$$f_k(x) = (2\pi)^{-p/2} |\Sigma_k|^{-1/2} \exp \left[-\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) \right]$$

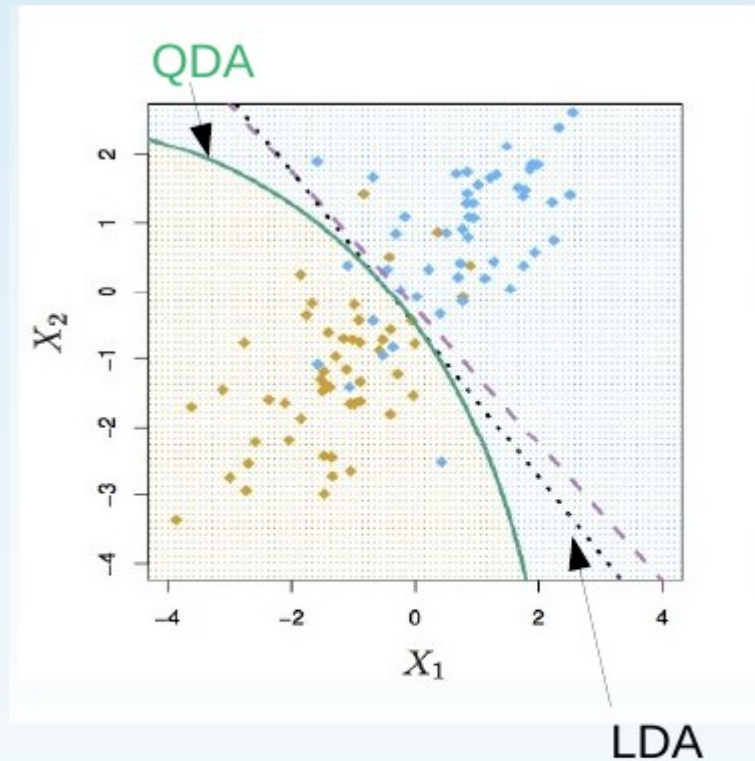
$$\begin{aligned} d_k(x) &= 2 \left[-\log f_k(x) - \log \pi_k \right] - \text{constant} \\ &= (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \log |\Sigma_k| - 2 \log \pi_k \end{aligned}$$

LDA versus QDA?

LDA versus QDA?

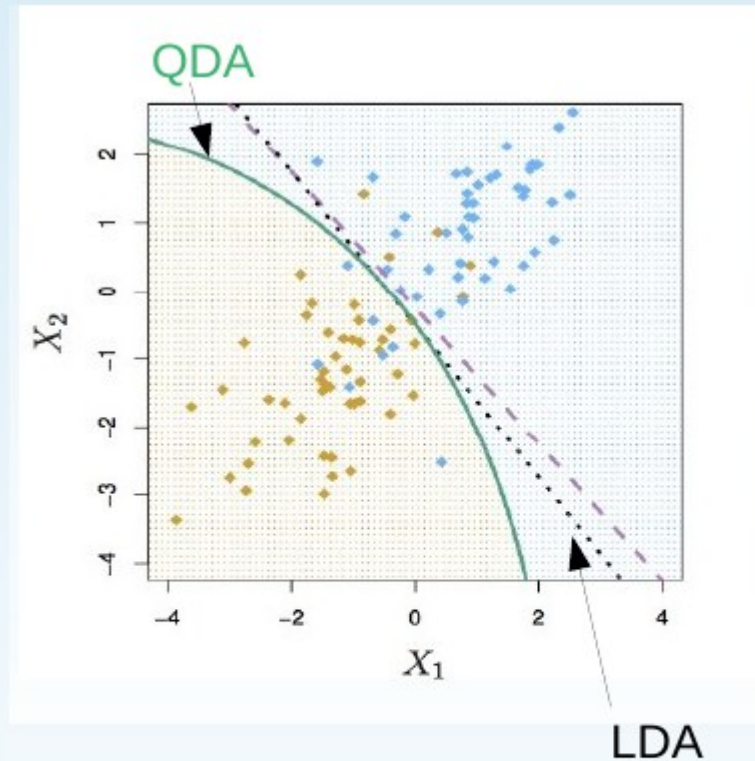
- QDA permite variâncias diferentes entre classes, os contornos resultantes são quadráticos
- Qual é melhor: LDA ou QDA?
 - QDA será melhor quando as variâncias forem muito diferentes entre as classes e tivermos observações suficientes para estimar as variâncias de forma acurada
 - LDA será melhor quando as variâncias forem similares entre classes e não tivermos dados suficientes para estimar de forma acurada as variâncias

LDA versus QDA?

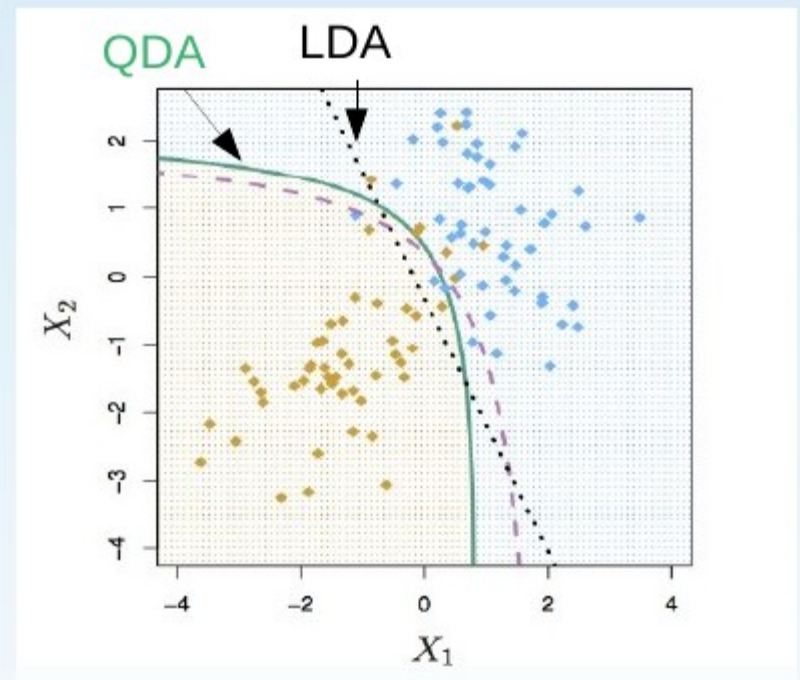


Variâncias semelhantes → LDA melhor

LDA versus QDA?



Variâncias semelhantes → LDA melhor



Variâncias diferentes → QDA melhor

Regressão Logística versus LDA?

- **Similaridade:** Ambas, Regressão Logística e LDA produzem contornos lineares.
- **Diferença:** LDA assume que as observações são retiradas de uma distribuição normal com variância comum em cada classe, enquanto que Regressão Logística não assume isso. LDA será melhor que Regressão Logística se a premissa de normalidade for satisfeita, caso contrário Regressão Logística será melhor.

Exemplo de projeto em python para rodar/estudar

Estudem, vejam resultados, formatos, avaliem...

Notebook no Kaggle

<https://www.kaggle.com/code/raissaid/classification-logistic-regression-lda-and-qda/notebook>

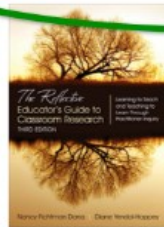
Classificação, comparando reg.logística, lda e qda, na predição de quem (homen ou mulher, realizará um click na propaganda disponibilizada)

Varáveis do dataset:

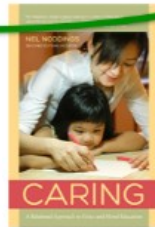
Daily Time Spent on Site, Age, Area Income, Daily Internet Usage, Ad Topic Line, City, Male, Country, Timestamp, Clicked on Ad

Exemplo de projeto “Sistemas de Recomendação”

Customers who bought this item also bought



The Reflective Educator's
Guide to Classroom
Research: Learning to...
Nancy Fichtman Dana
★★★★★ 1
Kindle Edition
CDN\$ 30.98



Caring: A Relational
Approach to Ethics and
Moral Education
Nel Noddings
Kindle Edition
CDN\$ 33.46

amazon

LinkedIn

Jobs you may be interested in

Any location · Any industry · 0 to 10,000+



Example: Recommender Systems

- A user goes to Amazon to buy products.
- Amazon has some information about the user. They also have information about other users buying similar products.
- What should they recommend to the user, so that they buy more products?
- There's no “right” answer (no label).
- The whole idea is to understand user behavior in order to recommend them products they are likely to consume.

Back to top

Exemplo de projeto “Sistemas de Recomendação”

Why should we care about recommendation systems?

- Almost everything we buy or consume today is in some way or the other influenced by recommendation systems.
 - Music (Spotify), videos (YouTube), news, books and products (Amazon), movies (Netflix), jokes, restaurants, dating , friends (Facebook), professional connections (LinkedIn)
- Recommendation systems are at the core of the success of many companies such as Amazon and [Netflix](#).



- Recommendation systems are often presented as powerful tools that significantly **reduce the effort users need to put in finding items**, effectively mitigating the problem of information overload.
- This is more or less true in many contexts. Consider, for instance, the experience of shopping an umbrella on Amazon without the help of recommendations or any specific ranking of products.
- In the absence of a recommendation system, users would be faced with the daunting task of sifting through thousands of available products to find the one that best suits their needs.

Exemplo de projeto “Sistemas de Recomendação”

Data and main approaches

What kind of data we need to build recommendation systems?

- Customer purchase history data (We worked with it last week.)
- **User-item interactions** (e.g., ratings or clicks) (most common)
- **Features related to items or users**

Exemplo de projeto “Sistemas de Recomendação”

Main approaches

- Collaborative filtering
 - “Unsupervised” learning
 - We only have labels y_{ij} (rating of user i for item j).
 - We learn latent features.
- **Content-based recommenders** (today's focus)
 - Supervised learning
 - Extract features x_i of users and/or items building a model to predict rating y_i given x_i .
 - Apply model to predict for new users/items.
- Hybrid
 - Combining collaborative filtering with content-based filtering

Exemplo de projeto “Sistemas de Recomendação”

Recommender systems problem












Problem formulation

- Most often the data for recommender systems come from Back to top **interactions** between a set of items and a set of users.
- We have two entities: N **users** and M **items**.
- **Users** are consumers.
- **Items** are the products or services offered.
 - E.g., movies (Netflix), books (Amazon), songs (spotify), people (tinder)
- A **utility matrix** is the matrix that captures **interactions** between N **users** and M **items**.
- The interaction may come in different forms:

Exemplo de projeto “Sistemas de Recomendação”

◦ ratings, clicks, purchases

- Below is a toy utility matrix. Here $N = 6$ and $M = 5$.
- Each entry y_{ij} (i^{th} row and j^{th} column) denotes the rating given by the user i to item j .
- We represent users in terms of items and items in terms of users.

						
		Item 1	Item 2	Item 3	Item 4	Item 5
	User 1	?	?	2	?	3
	User 2	3	?	?	?	?
	User 3	?	5	4	?	5
	User 4	?	?	?	?	?
	User 5	?	?	?	5	?
	User 6	?	5	4	3	?

- The utility matrix is very sparse because usually users only interact with a few items.
- For example:
 - all Netflix users will have rated only a small percentage of content available on Netflix
 - all amazon clients will have rated only a small fraction among all items available on Amazon

What do we predict?

Given a utility matrix of N users and M items, **complete the utility matrix**. In other words, **predict missing values in the matrix**.

- Once we have predicted ratings, we can recommend items to users they are likely to rate higher.
- Note: rating prediction \neq Classification or regression

Exemplo de projeto “Sistemas de Recomendação”







Content-based filtering

- What if a new item or a new user shows up?
 - You won't have any ratings information for that item or user
- Content-based filtering is suitable to predict ratings for new items and new users.
- Content-based filtering is a **supervised machine learning** approach to recommender systems.
- In KNN imputation (an example of collaborative filtering) we assumed that we only have ratings data.
- Usually, there is some information available about items and users.
[Back to top](#)
- Examples
 - Netflix can describe movies as action, romance, comedy, documentaries.
 - Netflix has some demographic and preference information on users.
 - Amazon could describe books according to topics: math, languages, history.
 - Tinder could describe people according to age, location, employment.
- Can we use this information to predict ratings in the utility matrix?
 - Yes! Using content-based filtering!




Exemplo de projeto “Sistemas de Recomendação”

In content-based filtering,

- We assume that we are given item or user feature.
- Given movie information, for instance, we **create user profile for each user**.
- We treat ratings prediction problem as **a set of regression problems** and build regression model for each user.
- Once we have trained regression models for each user, we **complete the utility matrix by predicting ratings for each user** using their corresponding models.

							
Eva	?	?	?	5.0	?	5.0	1.0
Jim	?	?	4.0	5.0	?	5.0	?
Pat	5.0	5.0	?	?	4.0	?	?
Sam	5.0	5.0	?	?	5.0	1.0	4.0

Eva's profile

	Animation	Drama	Documentary	Rating
	0	0	1	5.0
	0	0	1	5.0
	1	1	0	1.0

Pat's profile

	Animation	Drama	Documentary	Rating
	1	1	0	5.0
	1	1	0	5.0
	0	0	1	1.0
...

[Back to top](#)

Projeto “Sistemas de Recomendação”

Leia com atenção estes 2 textos para uma prática na próxima semana.

<https://www.iteratorshq.com/blog/an-introduction-recommender-systems-9-easy-examples/>

<https://365datascience.com/tutorials/how-to-build-recommendation-system-in-python/>

Exercícios/Leitura

- Ler o artigo: A few useful things about machine learning (P.Domingos), *Communications of the ACM*, 55 (10), 78-87, 2012. (há versões livres no scholar.google.com)
- Ler o capítulo 19 do livro Russell & Norvig (*Artificial Intelligence (4a.ed)*)
OU
- Ler o capítulo 2 do livro do Alpaydin, E. (*Introduction to Machine Learning*)

Referências Bibliográficas

- Alpaydin, E. *Introduction to Machine Learning*, MIT Press, 2010.
- Bishop, C. *Pattern Recognition and Machine Learning*. Springer, 2006.
- James, G.; Witten, D.; Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning with applications in R*, Springer, 2014.
- Mitchell, T. *Machine Learning*. McGraw Hill, 1997.