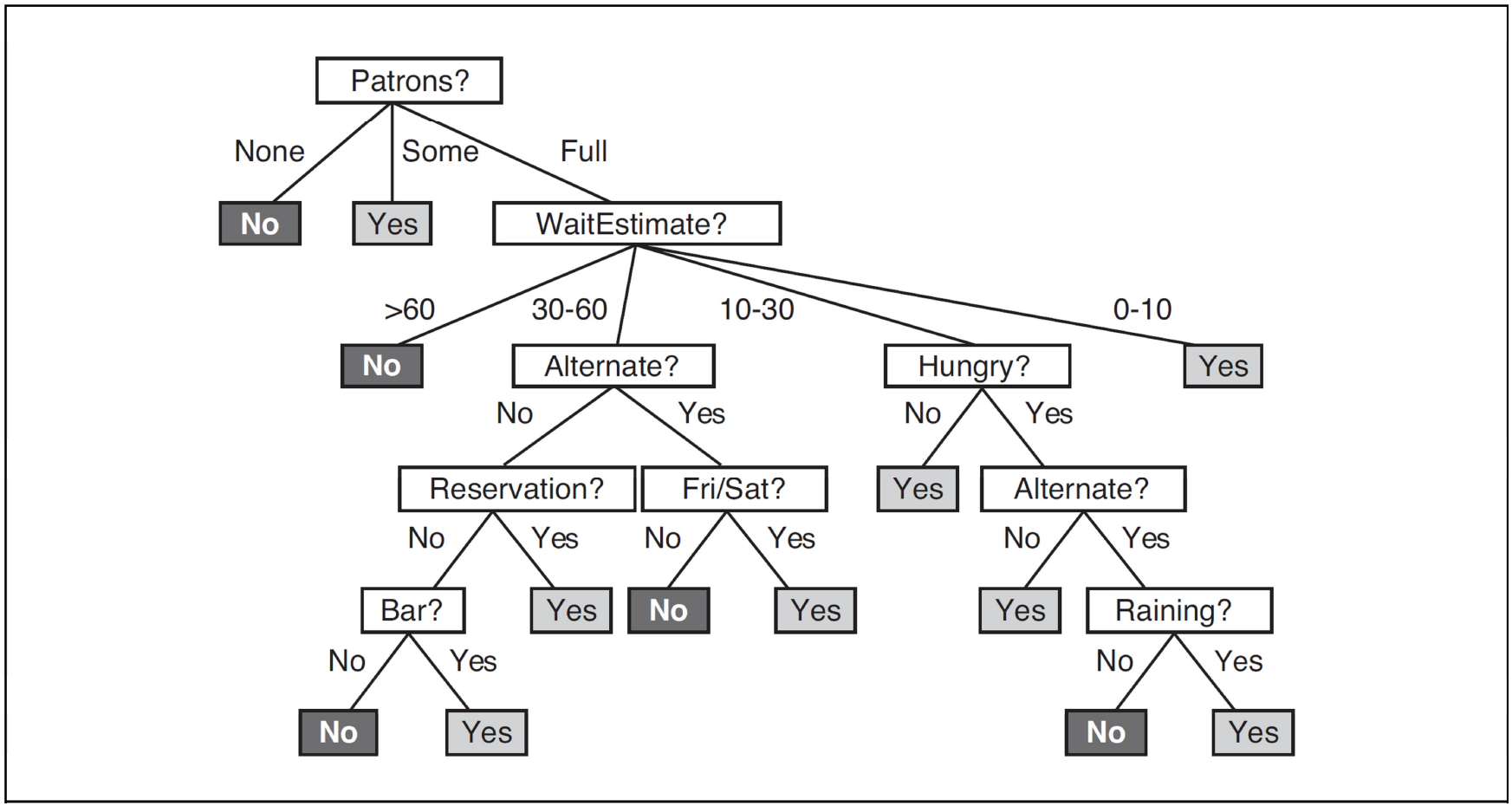
7-Aprendizagem com Exemplos

Árvores de decisão



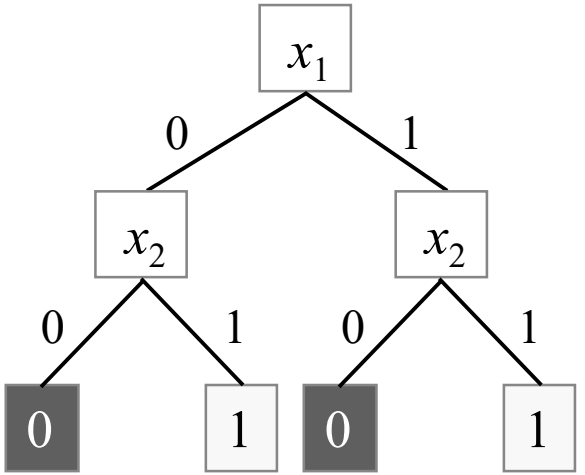
Árvore de decisão: representa uma função

entrada é um vetor de atributos (discretos e/ou contínuos)

saída é uma decisão (valor único)

caso particular: árvores booleanas

x_1	x_2	x_1 xor x_2
0	0	0
0	1	1
1	0	1
1	1	0



Expressividade de árvores booleanas

$Goal \Leftrightarrow (Path_1 \vee Path_2 \vee .....)$

Path : conjunção de testes (atributo-valor) no caminho

expressão na FNC $\rightarrow$ função (lógica proposicional) $\equiv$ árvore

nem sempre a representação mais concisa (e.g. função maioria)

h : hipótese é uma função booleana

H: espaço de hipóteses é o conjunto de todas funções booleanas

n atributos $\rightarrow$ 2^{2^n} funções

$n = 10 \rightarrow 2^{1024} = 10^{308}$ funções distintas !

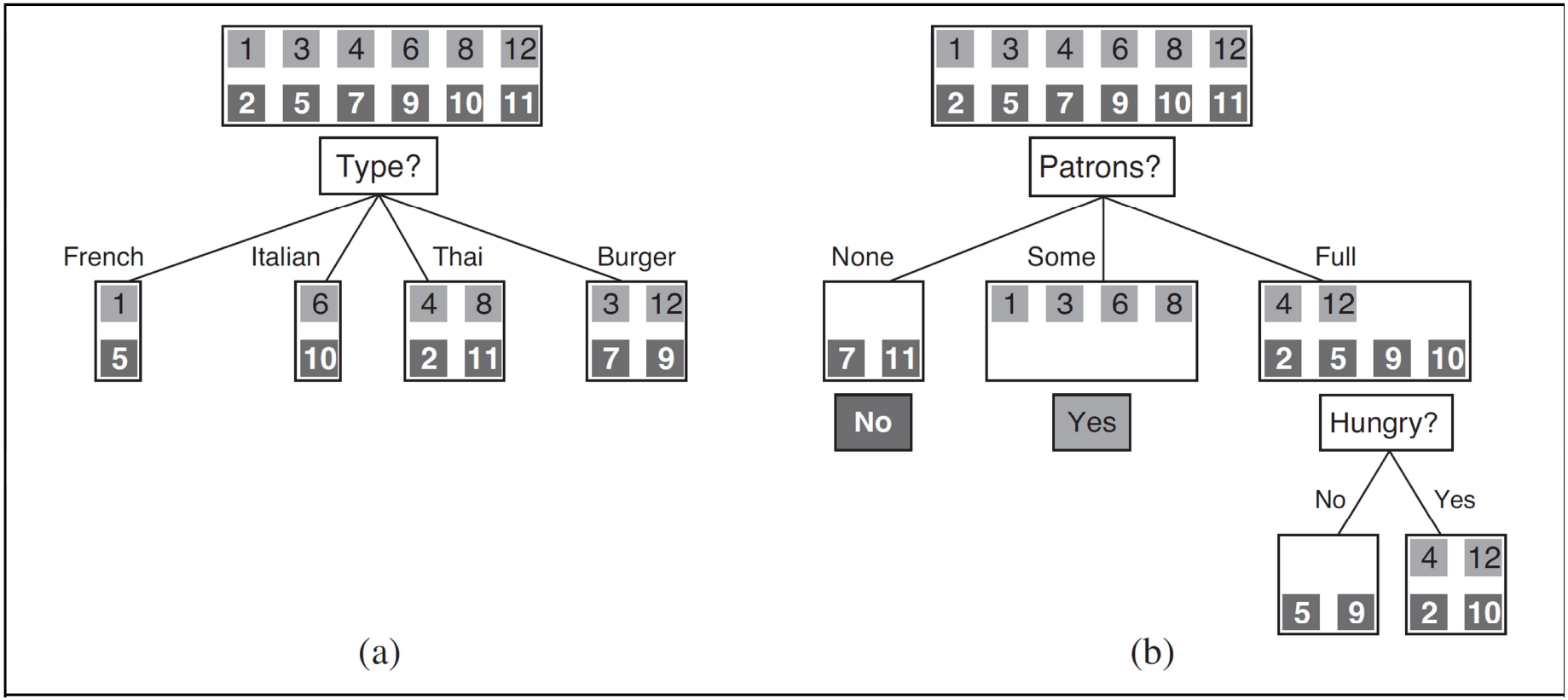
Indução de árvores booleanas

Dados de aprendizagem

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0 10*	$y_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30-60*	$y_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0-10*	$y_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10-30*	$y_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Same*	*$$*	*Yes*	*Yes*	*Italian*	*0-10*	$y_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0-10*	$y_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0-10*	$y_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10-30*	$y_{10} =$ *No*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0-10*	$y_{11} =$ *No*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30-60*	$y_{12} =$ *Yes*

Algoritmo de aprendizagem

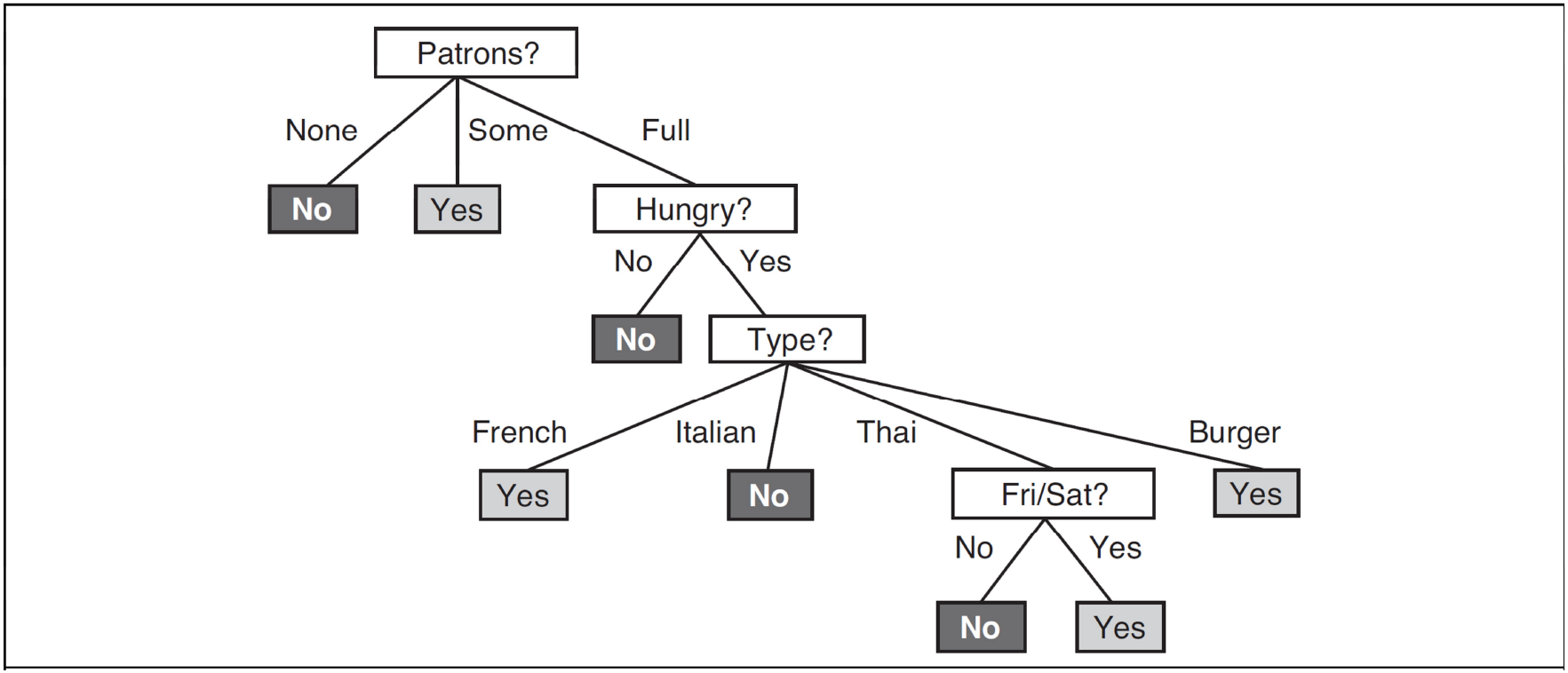
- testar o atributo mais importante primeiro, sempre
- atributo divide o conjunto de dados aprendizagem em subconjuntos
- cada subconjunto: dados para nova árvore (c/ um atributo a menos)
- quatro casos a considerar:
    1. exemplos restantes são todos + ou – : OK
    2. existem exemplos + e – : escolher melhor atributo para dividir
    3. não tem exemplos p/ combinação atributos: retornar valor default
    4. não tem mais atributos: retornar valor default (ruídos, etc.)



Algoritmo de aprendizagem de árvores

function DECISION_TREE_LEARNING (*examples, attributes, parent_examples*)
returns a tree

if *examples* is empty **then return** PLURARITY_VALUE(*parent_examples*)
else if all *examples* have a classification **then return** the classification
else if *attributes* is empty **then return** PLURARITY_VALUE(*parent_examples*)
else
$A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
tree ← a new decision tree with root test A
for each value v_k of A **do**
$exs \leftarrow \{e: e \in examples$ **and** $e.A = v_k\}$
subtree ← DECISION_TREE_LEARNING(*exs*, *attributesA*, *examples*)
add a branch to *tree* with label ($A = v_k$) and subtree *tree*
return *tree*

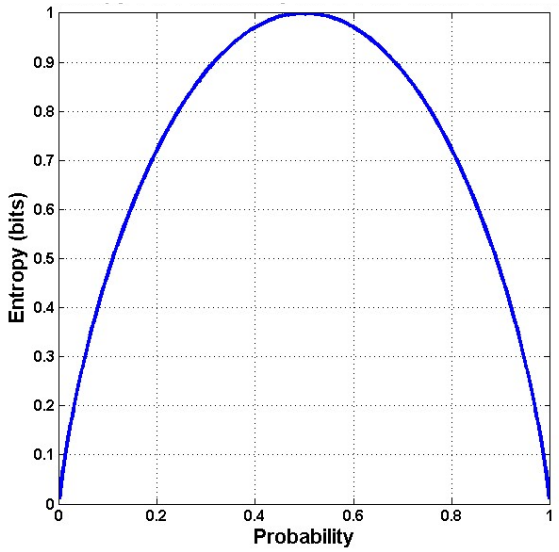


Escolha do atributo mais importante

V variável aleatória: valores v_k e $P(v_k)$

Entropia (Shannon and Weaver, 1949)

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$$



$$B(q) = -(q \log_2 q + (1-q) \log_2 (1-q))$$

Conjunto de dados de aprendizagem

p exemplos +

n exemplos –

$$H(Goal) = B\left(\frac{p}{p+n}\right)$$

Atributo A com d valores distintos

divide conjunto dados aprendizagem E em E_1,..., E_d subconjuntos

E_k cada com p_k exemplos + e n_k exemplos –

entropia associada a cada valor (ramo): $B(p_k/(p_k + n_k)$

probabilidade escolher exemplo de E com o k-ésimo valor: $(p_k + n_k)/(p + \mathrm{n})$

entropia depois de testar atributo A

$$Remainder\ (A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p + n} B\left(\frac{p_k}{p_k + n_k}\right)$$

Ganho de informação

$$Gain\ (A) = B\left(\frac{p}{p + n}\right) - Remainder\ (A)$$

Exemplo

$$Gain\ (Patrons) = B\left(\frac{6}{6+6}\right) - Remainder\ (Patrons)$$

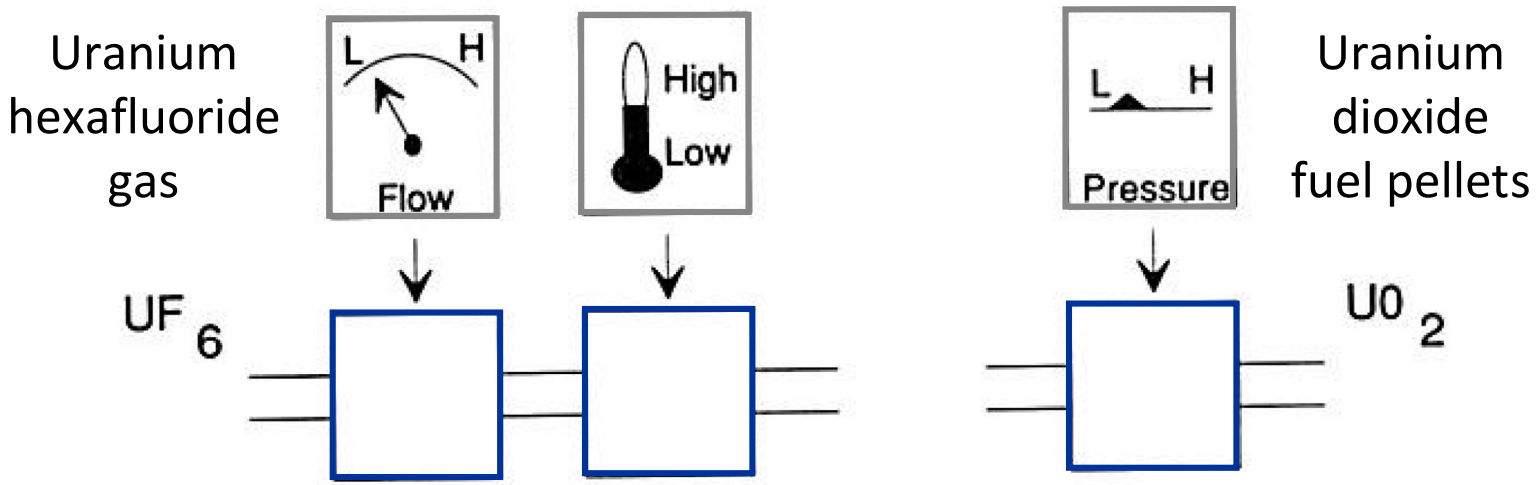
$$B\left(\frac{6}{6+6}\right) = B(0.5) = -(0.5\log_2 0.5 + (1-0.5)\log_2(1-0.5)) = 1$$

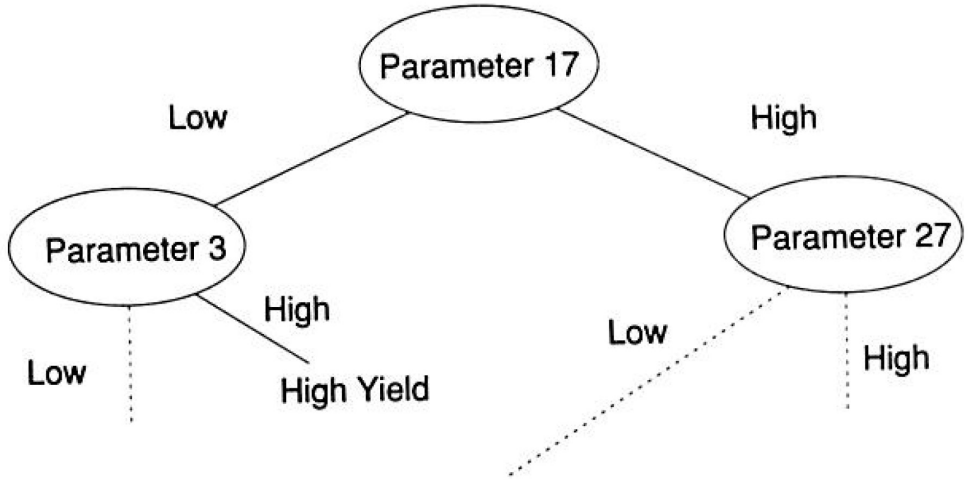
$$Remainder\ (Patrons) = \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right)\right] \approx 0.459 \text{ bits}$$

$$Gain\ (Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right)\right] \approx 0.541 \text{ bits}$$

$$Gain\ (Type) = 1 - \left[\frac{2}{12}B\left(\frac{1}{2}\right) + \frac{2}{12}B\left(\frac{1}{2}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) + \frac{4}{12}B\left(\frac{2}{4}\right)\right] \approx 0 \text{ bits}$$

Controle ótimo de processos





(P. Winston, 1993)

Regressão e classificação linear

h : funções afins

H: conjunto das funções afins

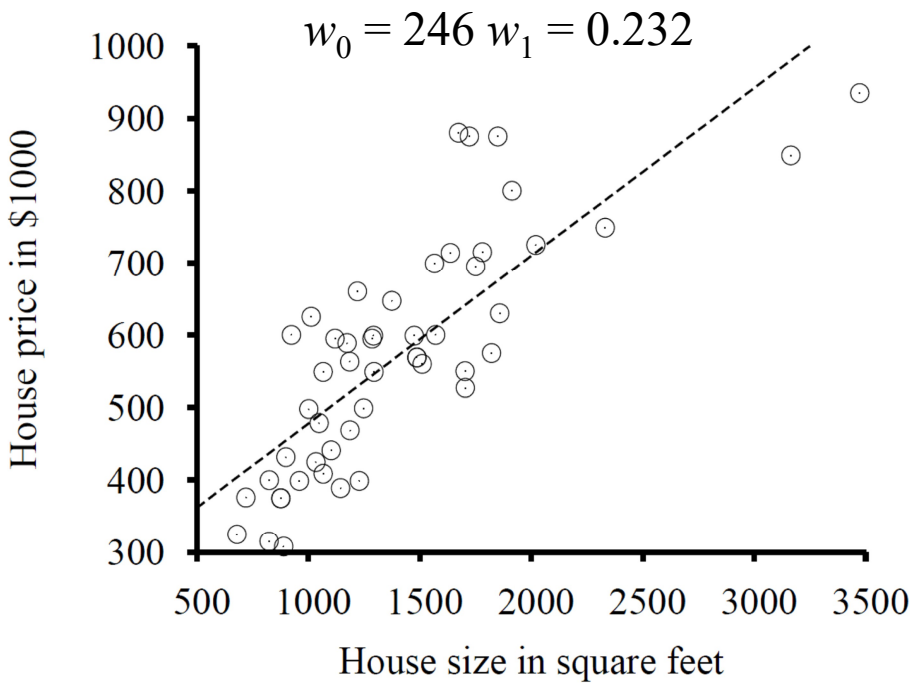
$h_{\mathbf{w}}(x) = w_0 + w_1 x \;\rightarrow\; H = \{\mathbf{w} \mid \mathbf{w} \in \mathrm{R}^{n+1}\}$

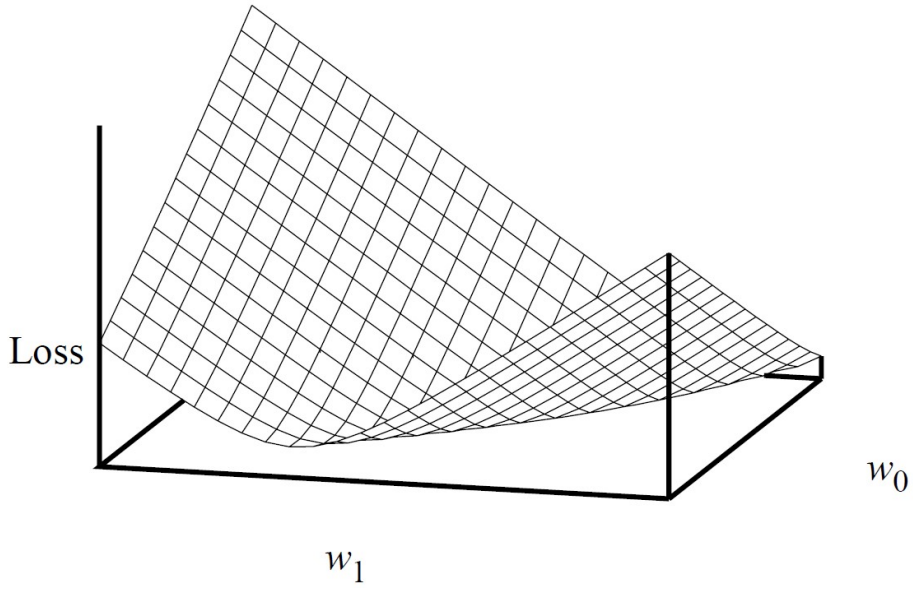
dados treinamento: $(x_1, y_1),\ldots, (x_j,y_j),\ldots,(x_N,y_N)$

determinar $\mathbf{w} = (w_0,w_1)$ que minimiza L_2 sobre todos dados de treinamento

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

$$\min_{\mathbf{w}} Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} (y_j - (w_0 + w_j x_j))^2$$





$$\mathbf{w}* = \arg\min_{\mathbf{w}} Loss(h_{\mathbf{w}})$$

$$\frac{\partial}{w_0} Loss(h_{\mathbf{w}}) = \frac{\partial}{w_0} \sum_{j=1}^{N} (y_j - (w_0 + w_j x_j))^2 = 0$$

$$\frac{\partial}{w_1} Loss(h_{\mathbf{w}}) = \frac{\partial}{w_1} \sum_{j=1}^{N} (y_j - (w_0 + w_j x_j))^2 = 0$$

Equações normais

$$\sum x_j y_j = \left(\sum x_j\right) w_0 + \left(\sum x_j^2\right) w_1 \qquad w_1* = \frac{N\left(\sum x_j y_j\right) - \left(\sum x_j\right)\left(\sum y_j\right)}{N\left(\sum x_j^2\right) - \left(\sum x_j\right)^2}$$

$$\sum y_j = N w_0 + \left(\sum x_j\right) w_1 \qquad w_0* = \frac{\sum y_j - w_1\left(\sum x_j\right)}{N}$$

Exemplo

x_j	1	3	4	6	8	9	11	14	$\sum x_j = 56$
y_j	1	2	4	4	5	7	8	9	$\sum y_j = 40$
x_j^2	1	9	16	36	64	81	121	196	$\sum x_j^2 = 524$
$x_j y_j$	1	6	16	24	40	63	88	126	$\sum x_j y_j = 364$
y_j^2	1	4	16	16	25	49	64	81	$\sum y_j^2 = 256$

$$w_1^* = \frac{N\left(\sum x_j y_j\right) - \left(\sum x_j\right)\left(\sum y_j\right)}{N\left(\sum x_j^2\right) - \left(\sum x_j\right)^2} = \frac{8(364) - (56)(40)}{8(524) - (56)^2} = \frac{7}{11} \approx 0.636$$

$$w_0^* = \frac{\sum y_j - w_1\left(\sum x_j\right)}{N} = \frac{40 - 7/11(56)}{8} = \frac{6}{11} \approx 0.545$$

Algoritmo do gradiente

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

$$\frac{\partial}{\partial w_i} Loss(h_{\mathbf{w}}) = \frac{\partial}{\partial w_i} \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2$$

$$\frac{\partial}{\partial w_i} Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j)) \frac{\partial (y_j - h_{\mathbf{w}}(x_j))}{\partial w_i}$$

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

batch gradient descent learning rule

Aproximação estocástica

$$\min_{\mathbf{w}} Loss(h_{\mathbf{w}}) = (y_j - h_{\mathbf{w}}(x_j))^2$$

$$\frac{\partial}{\partial w_i} Loss(h_{\mathbf{w}}) = \frac{\partial (y_j - h_{\mathbf{w}}(x_j))^2}{\partial w_i}$$

$$\frac{\partial}{\partial w_i} Loss(h_{\mathbf{w}}) = (y_j - h_{\mathbf{w}}(x_j)) \frac{\partial (y_j - h_{\mathbf{w}}(x_j))}{\partial w_i}$$

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

stochastic gradient descent learning rule

Exemplo regressão linear monovariável

Batch gradient descent

$$w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j))$$

$$w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) x_j$$

Aproximação estocástica

$$w_0 \leftarrow w_0 + \alpha (y_j - h_{\mathbf{w}}(x_j))$$

$$w_1 \leftarrow w_1 + \alpha (y_j - h_{\mathbf{w}}(x_j)) x_j$$

Regressão linear multivariável

$$h_{\mathbf{w}}(\mathbf{x}'_j) = w_0 + w_1 x_{j1} + w_2 x_{j2} + \cdots + w_n x_{jn}$$

$$\mathbf{x} = (x_{j0}, x_{j1}, \ldots, x_{jn})^{\mathrm{T}} \quad x_{j0} = 1$$

$$h_{\mathbf{w}}(\mathbf{x}_j) = \mathbf{w}.\mathbf{x} = \mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_i w_i x_i$$

$$\mathbf{w}* = \arg\min_{\mathbf{w}} \sum_j L_2(y_j, \mathbf{w}.\mathbf{x})$$

Solução analítica

$$\mathbf{w}* = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ & \vdots & & \\ x_{j1} & x_{j2} & \cdots & x_{jn} \\ & \vdots & & \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_j^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_N \end{bmatrix}$$

data matrix

Gradiente

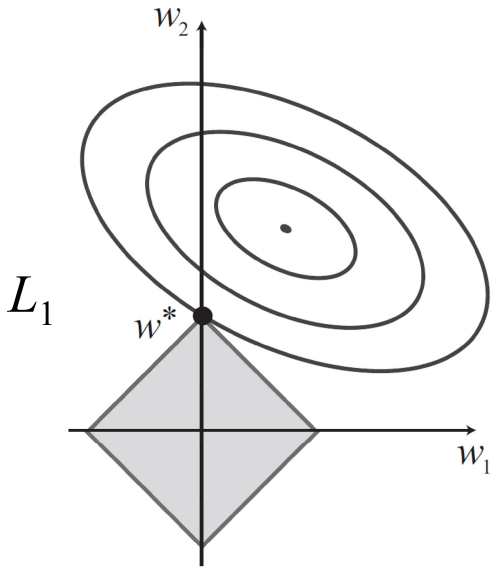
$$w_i \leftarrow w_i + \alpha \sum_j (y_j - h_{\mathbf{w}}(\mathbf{x}_j)) x_{ji}$$

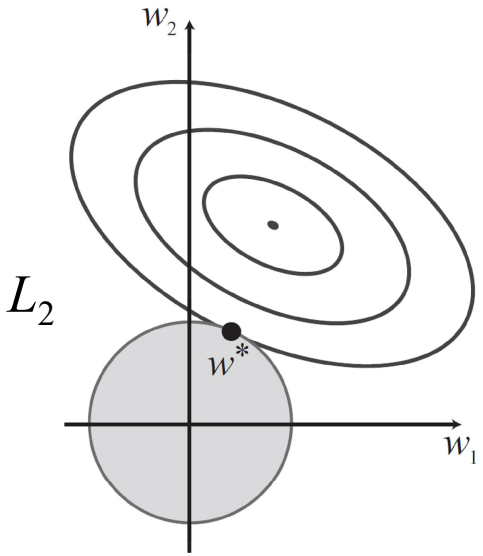
Regularização

ideia: evitar sobreajuste

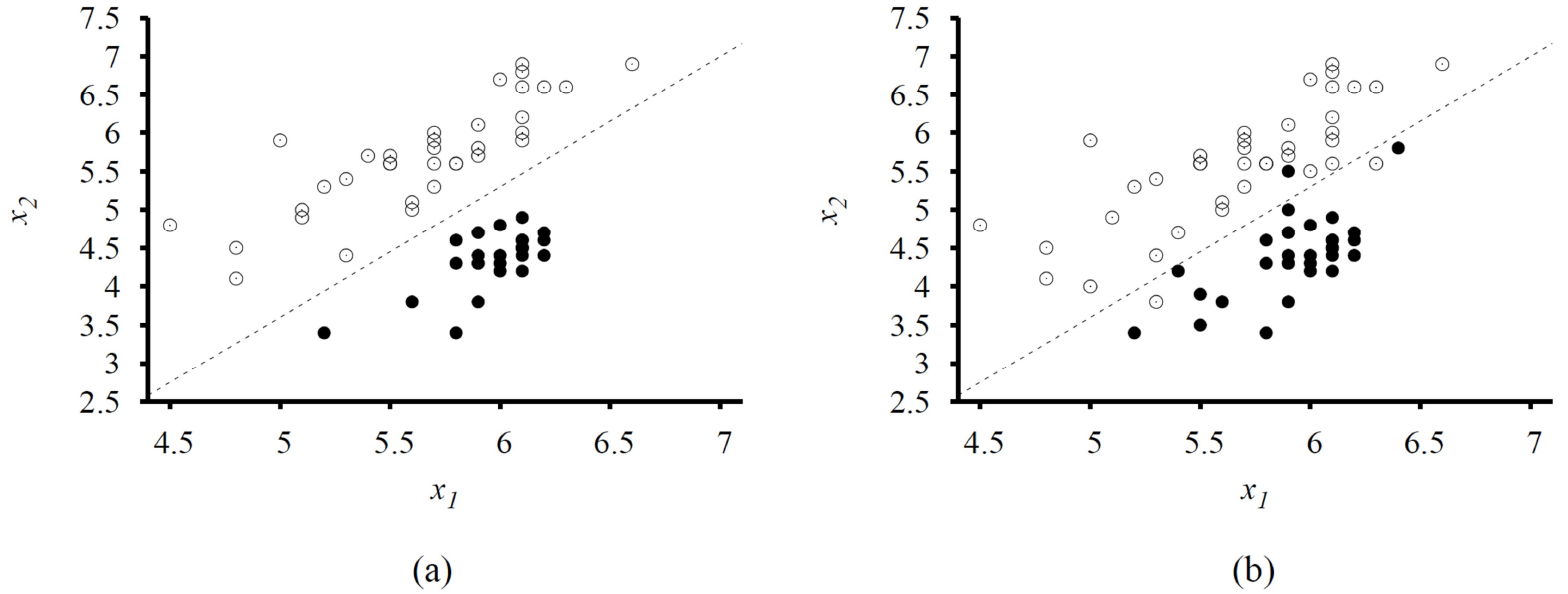
$$Cost(h_{\mathbf{w}}) = Loss(h_{\mathbf{w}}) + \lambda\ Complexity\ (h_{\mathbf{w}}) \quad \lambda > 0$$

$$Complexity\,(h_{\mathbf{w}}) = L_q(\mathbf{w}) = \sum_i |\, w_i \,|^q$$

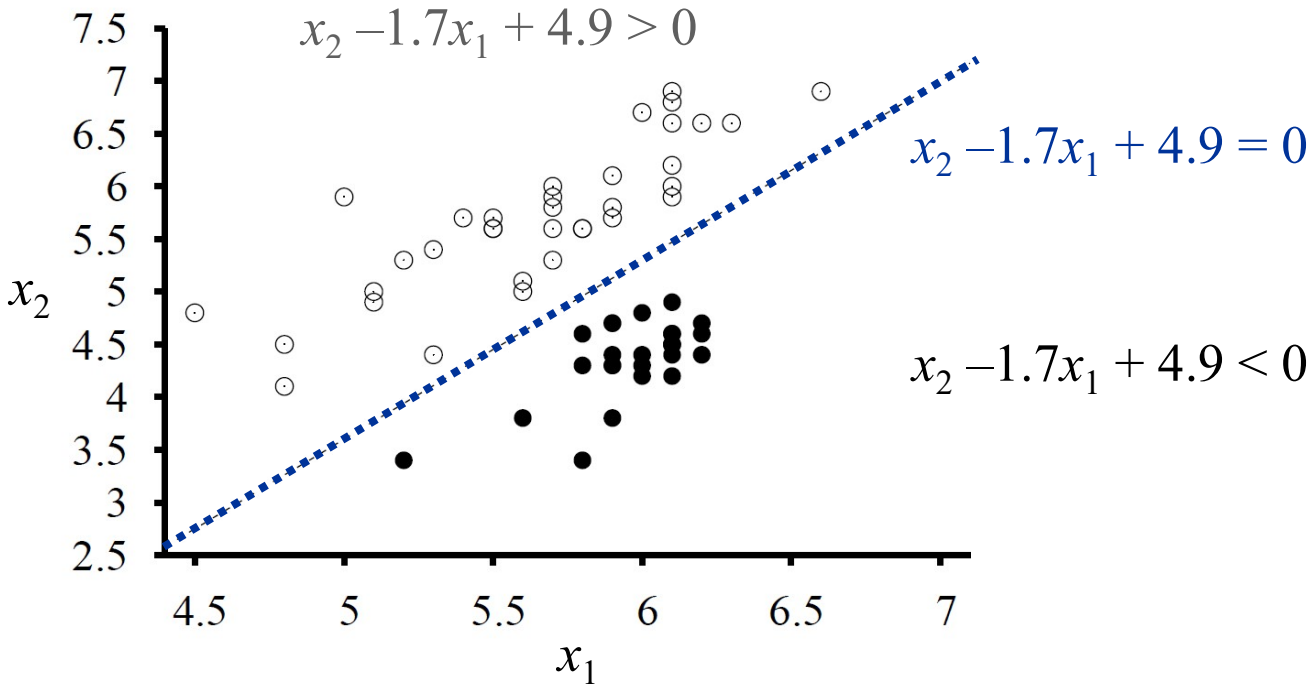




Classificação linear



(a) Plot of two seismic data parameters, body wave magnitude x_1 and surface wave magnitude x_2, for earthquakes (white circles) and nuclear explosions (black circles) occurring between 1982 and 1990 in Asia and the Middle East (?). Also shown is a decision boundary between the classes. (b) The same domain with more data points. The earthquakes and explosions are no longer linearly separable.



superfície de decisão (linear): $x_2 - 1.7x_1 + 4.9 = 0$

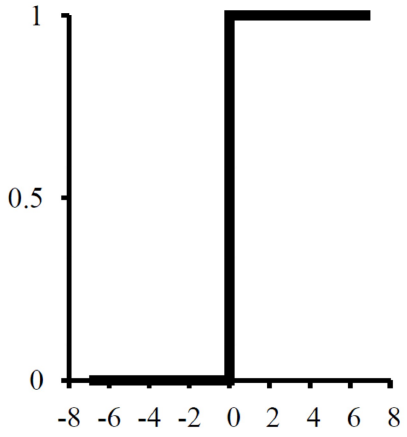
classe 1 = 0: explosão $x_2 - 1.7x_1 + 4.9 < 0$

classe 2 = 1: terremoto $x_2 - 1.7x_1 + 4.9 > 0$

Threshold

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 & \text{se } \mathbf{x}.\mathbf{w} \geq 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$h_{\mathbf{w}}(\mathbf{x}) : R^n \rightarrow \{0,1\}$$



dado o exemplo ($\mathbf{x}$, y)

$$w_i \leftarrow w_i + \alpha(y - h_{\mathbf{w}}(\mathbf{x}))x_i$$ perceptron learning rule

converge para uma solução se classes são linearmente separáveis

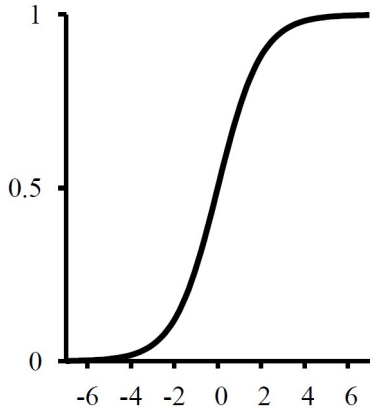
Regressão logística

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

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

$$h_{\mathbf{w}}(\mathbf{x}) : R^n \rightarrow [0,1]$$

$$\mathbf{w}* = \arg\min_{\mathbf{w}} Loss(h_{\mathbf{w}})$$



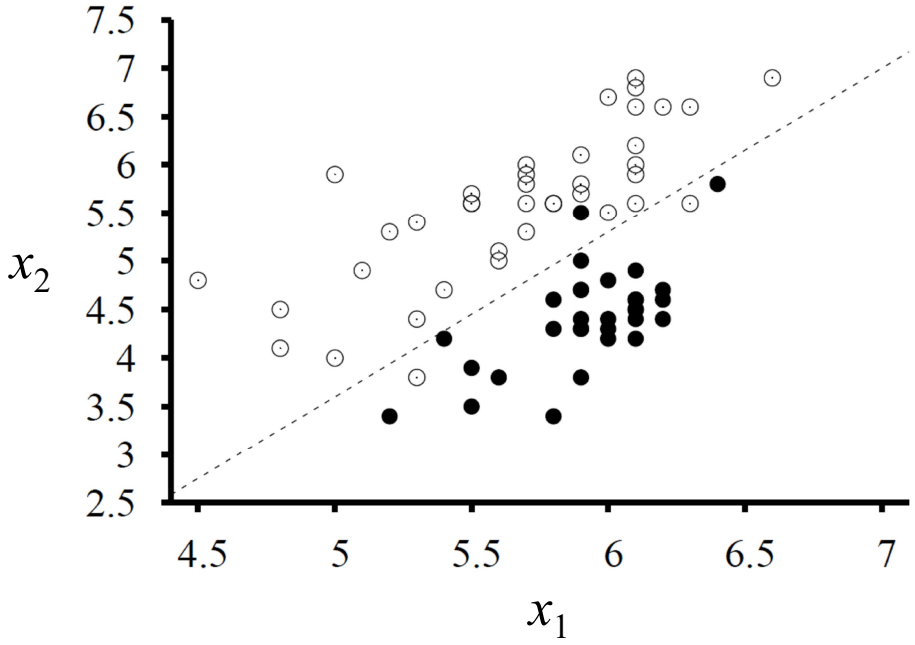
dado o exemplo $(\mathbf{x}, y)$

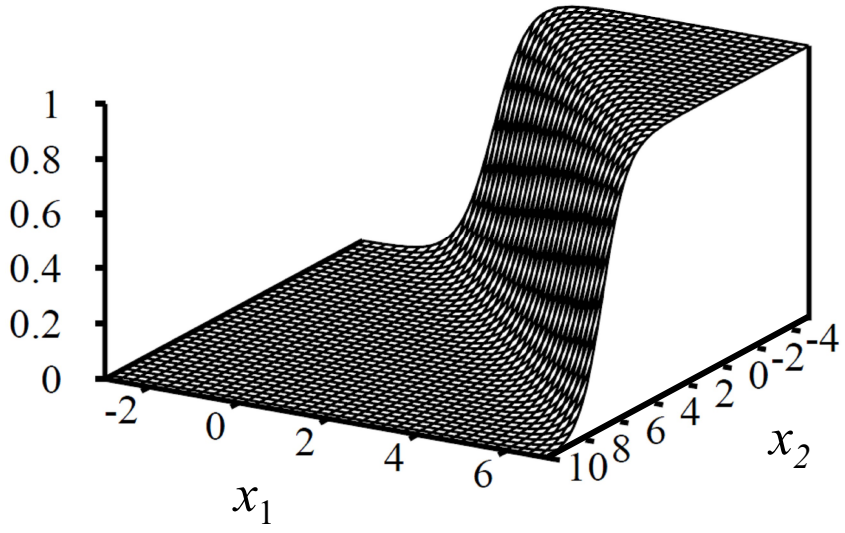
$$\min_{\mathbf{w}} Loss(h_{\mathbf{w}}) = (y_j - h_{\mathbf{w}}(x_j))^2$$

$$\frac{\partial Loss(h_{\mathbf{w}})}{\partial w_i} = -2(y - h_{\mathbf{w}}(\mathbf{x}))h'_{\mathbf{w}}(\mathbf{x})x_i$$

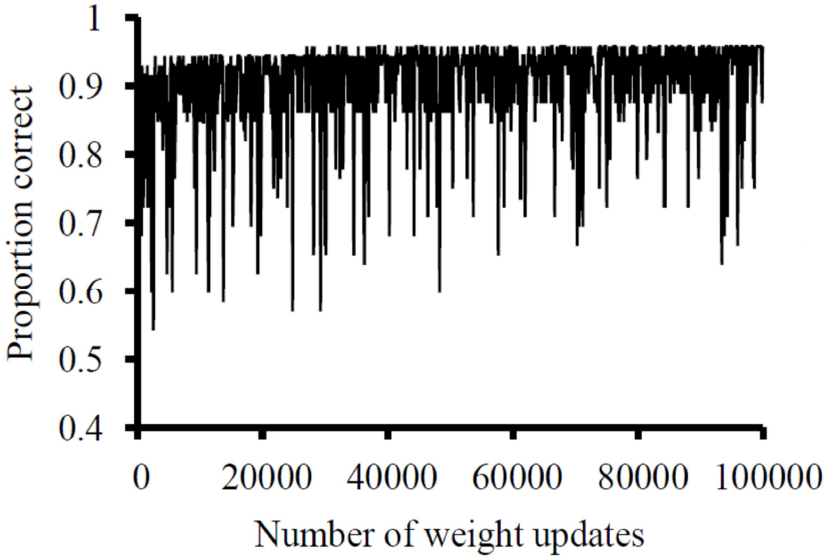
$$h'_{\mathbf{w}}(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x}))$$

$$w_i \leftarrow w_i + \alpha(y - h_{\mathbf{w}}(\mathbf{x}))h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x}))x_i$$ learning rule

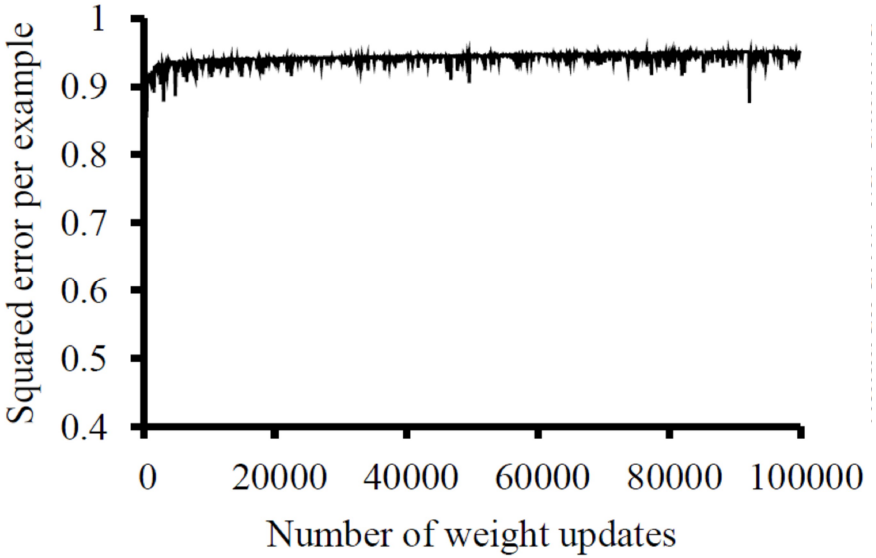




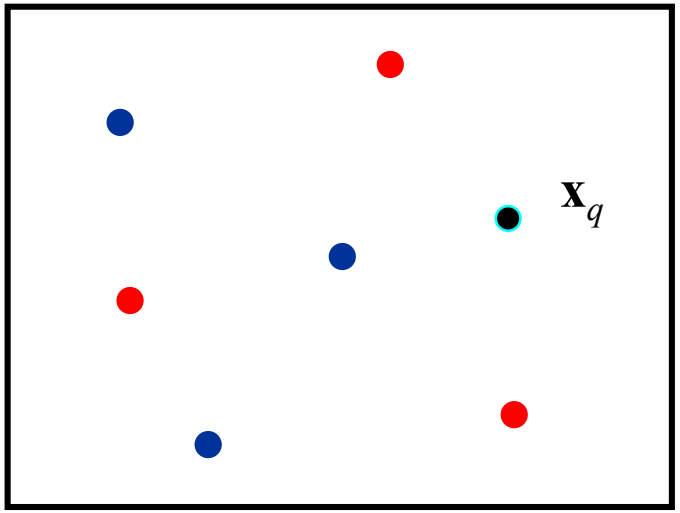
Threshold

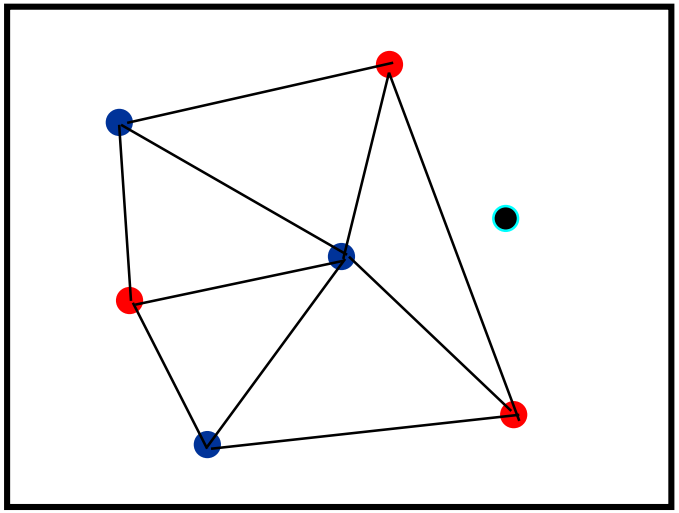


Logistic

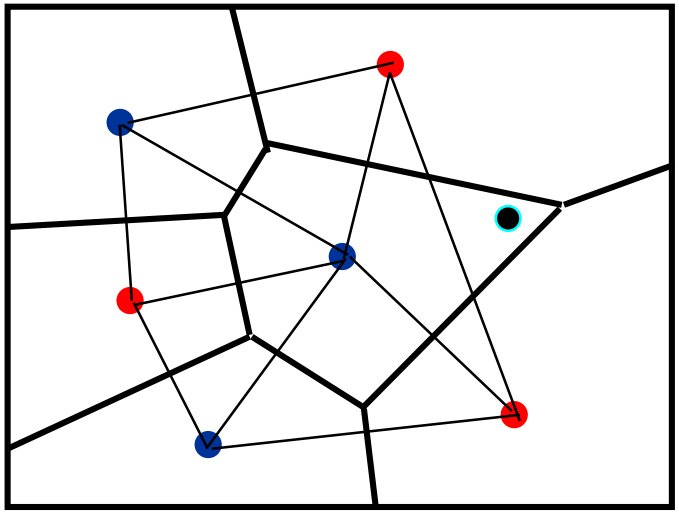


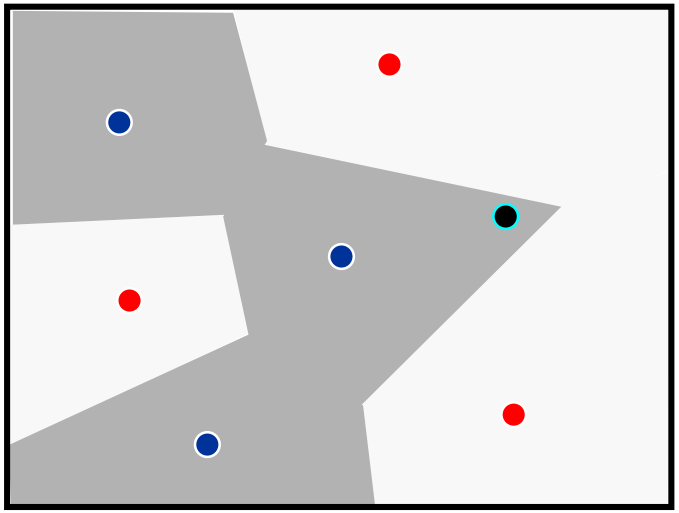
Vizinhos mais próximos





$\mathbf{x}_q$ ● ou ● ?





exemplos de treinamento $(\mathbf{x}, v_i)$, $v_i = f(\mathbf{x}_i)$

dado $\mathbf{x}_q$ encontrar os k exemplos mais próximos de $\mathbf{x}_q$

H: espaço de hipóteses não é explicito

$h(\mathbf{x})$: $R^n \rightarrow V$ implicitamente um diagrama de Voronoi

$V = \{v_1, v_2,...,v_s\}$ conjunto de rótulos/classes

$NN(k,\mathbf{x}_q)$: conjunto dos k vizinhos mais próximos de $\mathbf{x}_q$

k : número ímpar

treinamento rápido: rotulação de exemplos em uma lista

pode aprender superfícies/funções complexas de decisão

previsão pode ser lenta $O(nN)$

Aprendizagem *k NN*

Armazenar exemplos de treinamento ($\mathbf{x}, f(\mathbf{x}_i)$)

Dados numéricos (números reais): padronizar/normalizar

Sequential table lookup: $O(N)$

Árvore binária (k-d tree): $O(\log N)$

Hash table (local sensitive hash): $O(1)$

k-d trees: apropriada quando $N >>> n$ (2^n exemplos mínimo)

Previsão com k NN

Classificação

determinar $NN(k, \mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$

$$h(\mathbf{x}_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(\mathbf{x}_i))$$

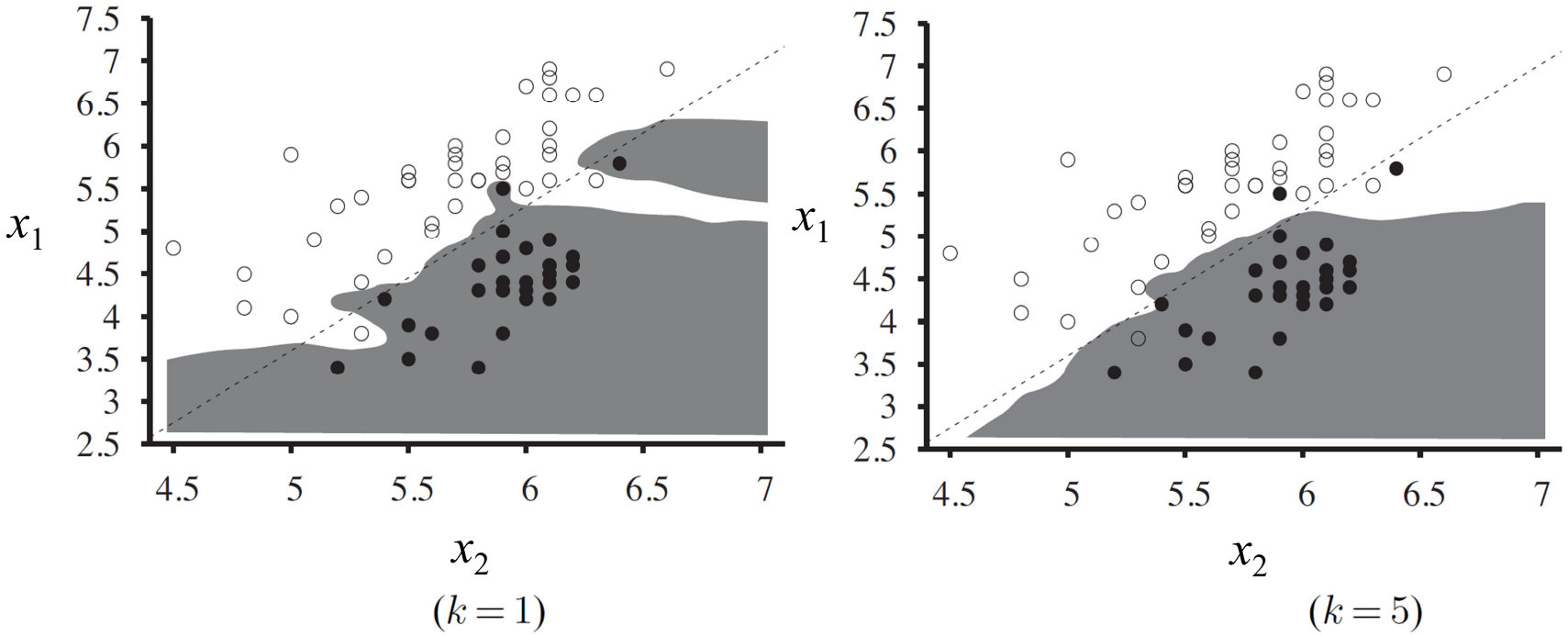
$\delta(a,b) = 1$ se $a = b$ $\delta(a,b) = 0$ caso contrário

Regressão

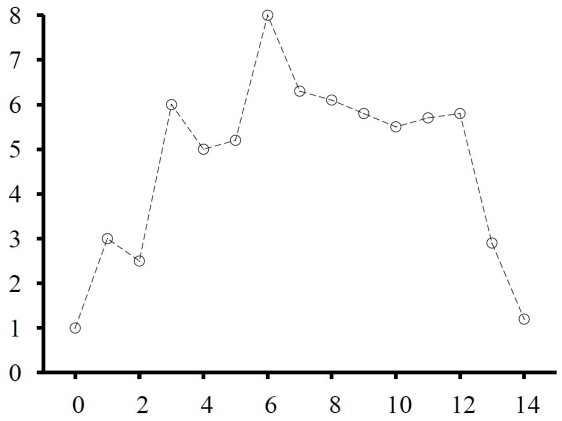
determinar $NN(k, \mathbf{x}_q) = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$

$$h(\mathbf{x}_q) \leftarrow \frac{\sum_{i=1}^{k} f(\mathbf{x}_i)}{k}$$

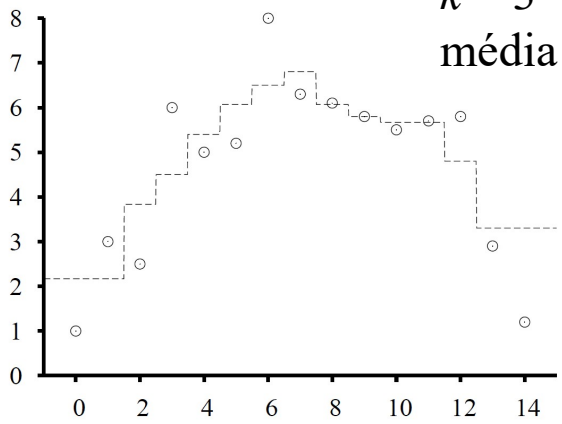
Classificação



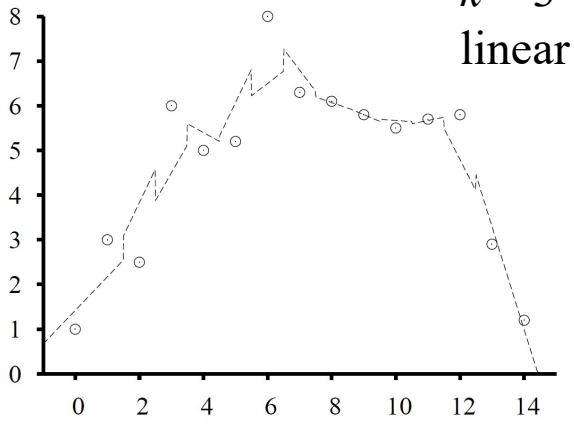
Regressão



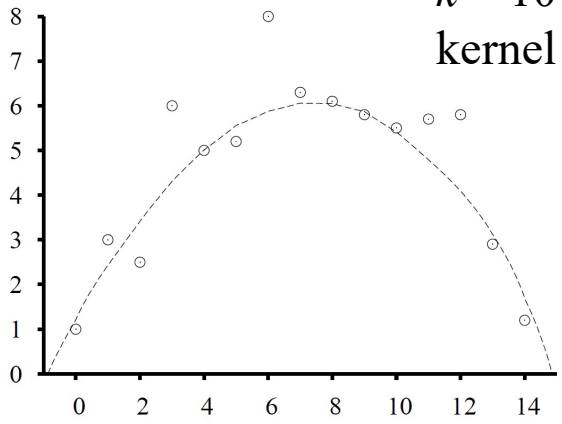
(a)



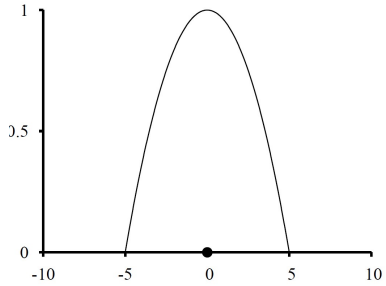
(b)



(c)

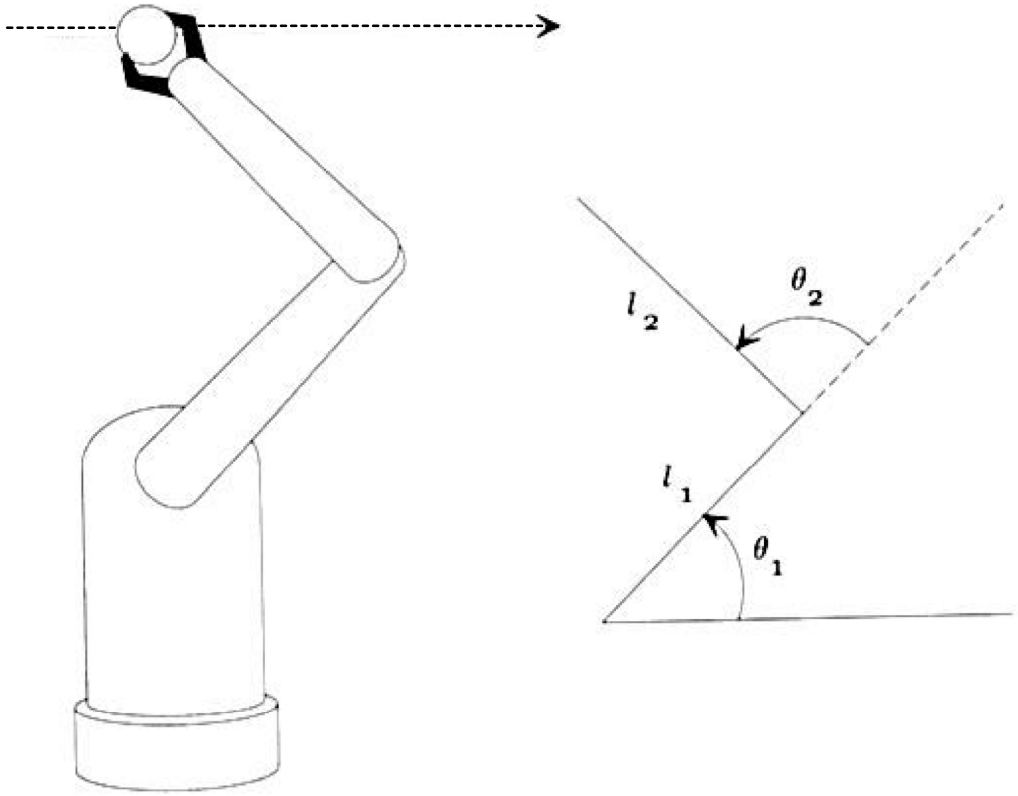


(d)



$\mathcal{K}(d) = \max(0, 1 - (2|x|/k)^2)$

Controle de robôs industriais



Modelo cinemático

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 sin\,\theta_1 + l_2 sin\,(\theta_1 + \theta_2)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2 sin\,\theta_2}{l_1 + l_2 \cos\theta_2}\right)$$

$$\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

(P. Winston, 1993)

Modelo dinâmico

$$\tau_1 = \ddot{\theta}_1 (I_1 + I_2 + m_2 l_1 l_2 \cos\theta_2 + \frac{m_1 l_1^2 + m_2 l_2^2}{4} + m_2 l_1^2)$$

$$+ \ddot{\theta}_2 (I_2 + \frac{m_2 l_2^2}{4} + \frac{m_2 l_1 l_2}{2} \cos\theta_2)$$

$$- \dot{\theta}_2^2 m_2 l_1 l_2 sin\theta_2$$

$$- \dot{\theta}_1 \dot{\theta}_2 m_2 l_1 l_2 sin\theta_2$$

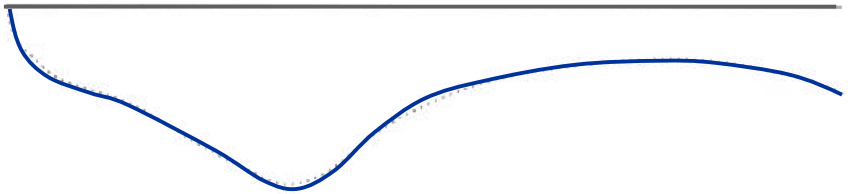
$$\tau_2 = \ddot{\theta}_1 (I_2 + \frac{m_2 l_1 l_2 \cos\theta_2}{2} + \frac{m_2 l_2^2}{4})$$

$$+ \ddot{\theta}_2 (I_2 + \frac{m_2 l_2^2}{4})$$

$$- \dot{\theta}_1^2 \frac{m_2 l_1 l_2}{2} sin\theta_2$$

τ_1	τ_2	θ_1	θ_2	$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_1^2$	$\dot{\theta}_2^2$	$\dot{\theta}_1\dot{\theta}_2$	$\ddot{\theta}_1^2$	$\ddot{\theta}_2^2$

Tabela inicial



Uma iteração

Três iterações

Medidas de proximidade

Distância de Minkowski

$$L^p(\mathbf{x}_i, \mathbf{x}_q) = \left(\sum_i | x_{ji} - x_{qi} |^p \right)^{1/p}$$

$p = 1$ Manhattan
$p = 2$ euclidiana

$$L^H(\mathbf{x}_i, \mathbf{x}_q) = \sum_i | x_{ji} - x_{qi} |$$

$p = 1$ Hamming ($x_{ji}, x_{qi} \in \{0,1\}$)

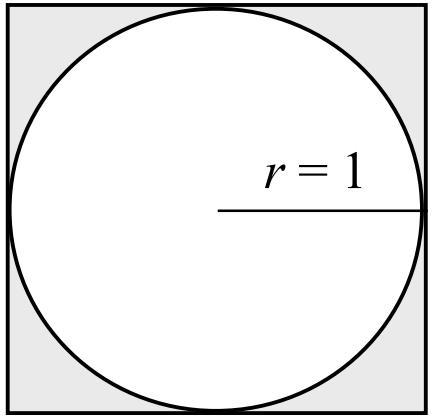
$$L^\infty(\mathbf{x}_i, \mathbf{x}_q) = \max_i | x_{ji} - x_{qi} |$$

$$L^{-\infty}(\mathbf{x}_i, \mathbf{x}_q) = \min_i | x_{ji} - x_{qi} |$$

$p \to \pm\infty$ Chebyshev

Espaços de grande dimensão

Fração dos pontos em um cubo que está fora da esfera inscrita raio unitário?



$$V_e = \frac{\pi^{\frac{n}{2}}}{\frac{1}{2}\Gamma\left(\frac{n}{2}\right)}$$

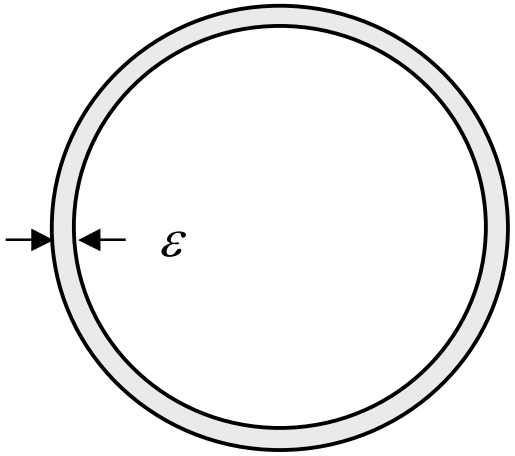
$$\frac{V_s - V_e}{V_s} = 1 - \frac{V_e}{V_s}$$

$$\lim_{n\to\infty} \frac{V_e}{V_s} = 0$$

$$\lim_{n\to\infty} \frac{V_s - V_e}{V_s} = 1 \quad !$$

$$\Gamma(1) = \Gamma(2) = 1,\ \Gamma(1/2) = \sqrt{\pi},\ \Gamma(3/2) = \frac{1}{2}\sqrt{\pi}$$

Fração do volume entre esferas de raio unitário e $(1-\varepsilon)$?



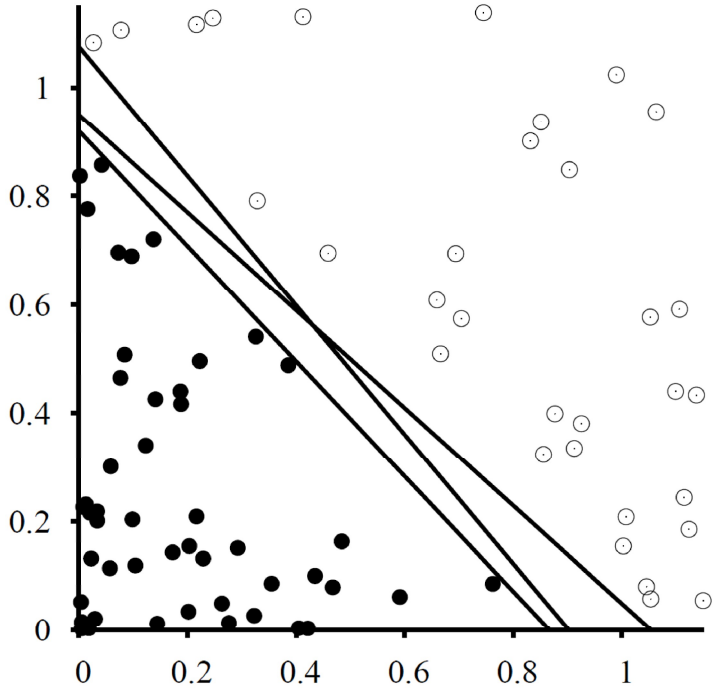
$$V_e = \frac{\pi^{\frac{n}{2}}}{\frac{1}{2}\Gamma\left(\frac{n}{2}\right)} \qquad V_\varepsilon = (1-\varepsilon)^n V_e$$

$$\lim_{n\to\infty} \frac{V_e - V_\varepsilon}{V_e} = \lim_{n\to\infty}\left(1 - \frac{V_\varepsilon}{V_e}\right) = 1 - \lim_{n\to\infty}(1-\varepsilon)^n$$

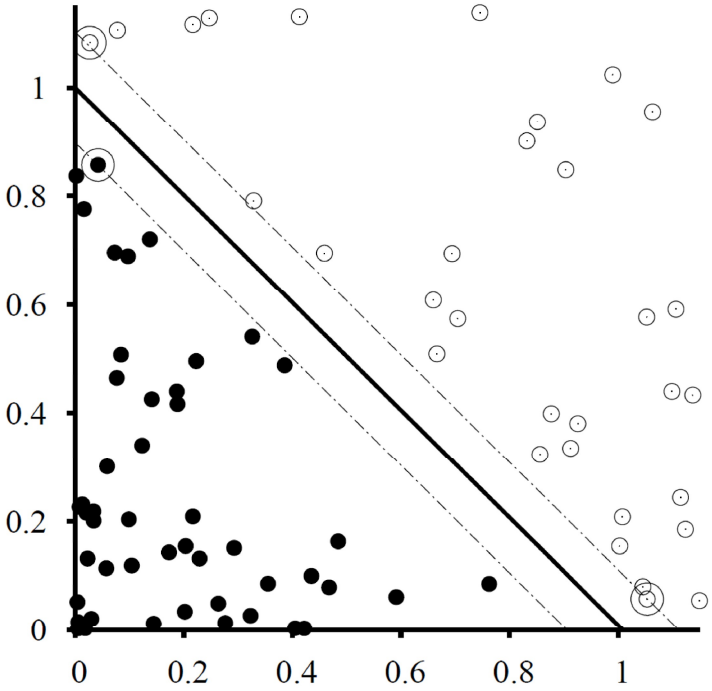
$$\lim_{n\to\infty}(1-\varepsilon)^n = 0 \qquad \lim_{n\to\infty} \frac{V_e - V_\varepsilon}{V_e} = 1 \qquad !$$

maior parte do volume esfera r < 1 está em uma casca de espessura $O(1–r/n)$

Máquinas de vetores de suporte



(a)



(b)

Conveniente quando não se tem conhecimento especializado

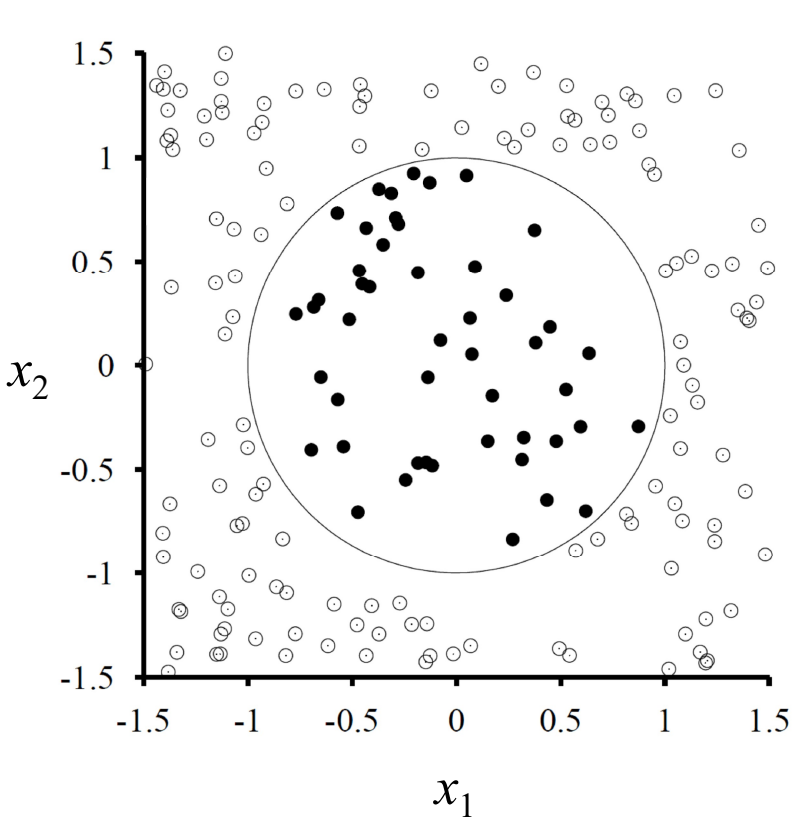
Constrói separador de máxima margem

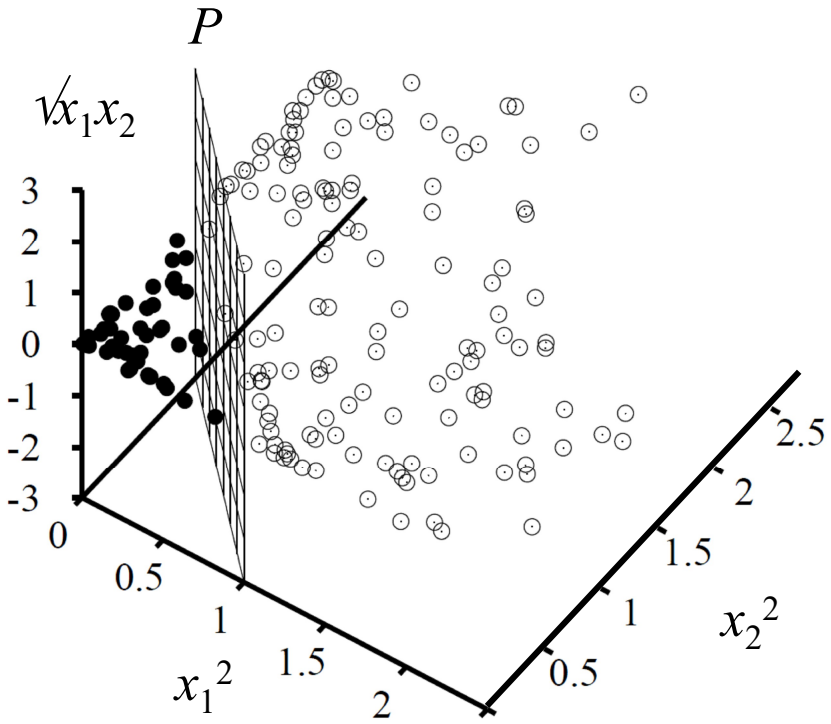
Incorpora dados em espaço de grande dimensão (*kernel trick*)

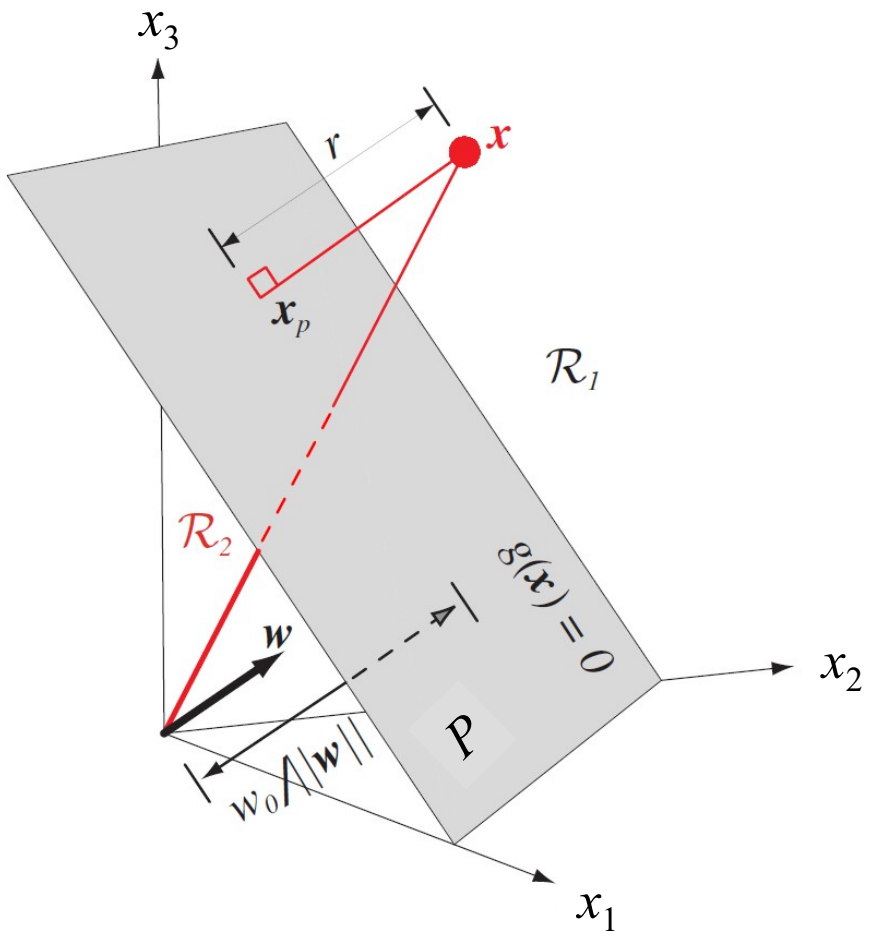
Linear em grande dimensão é não linear no espaço original

H é expandido além das representações lineares

Não paramétrico, na prática combina vantagens de ambos


1.5
1
0.5
0
-0.5
-1
-1.5
x_2
-1.5
-1
-0.5
0
0.5
1
1.5
x_1


P
$\sqrt{x_1 x_2}$
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
x_1^2
0.5
1
1.5
2
2.5
x_2^2



$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}$$

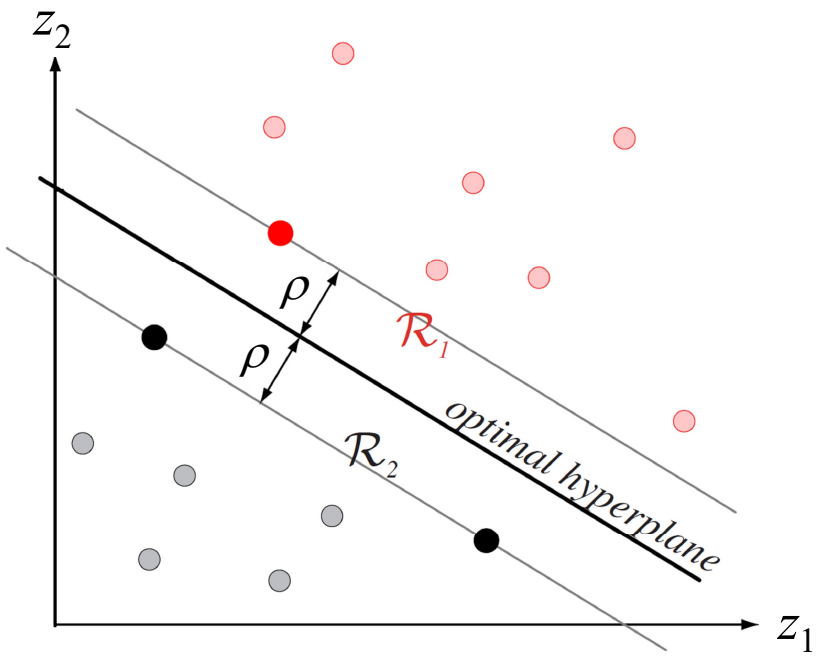
Exemplos $(\mathbf{z}_j, y_j)$ $\mathbf{z}_j = F(\mathbf{x}_j),\ y_j \in \{+1,-1\}\quad j = 1,\ldots,N$

$g(\mathbf{z}) = \mathbf{w}^{\mathrm{T}}\mathbf{z} + b$

$y_j g(\mathbf{z}) \geq 1 \quad j = 1,\ldots,N$

Margem $\dfrac{y_j g(\mathbf{z}_j)}{\|\mathbf{w}\|} \geq \rho$

$\rho\|\mathbf{w}\| = 1$



Problema primal

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|$$

$$\text{sa} \quad y_j(\mathbf{w}^{\mathrm{T}}\mathbf{z}_j + b) \geq 1, \quad j = 1,\ldots,N$$

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\| + \sum_{j=1}^{N} \alpha_j (1 - y_j(\mathbf{w}^{\mathrm{T}}\mathbf{z}_j + b))$$

$$\mathbf{w} = \sum_{j=1}^{N} \alpha_j y_j \mathbf{z}_j \qquad \sum_{j=1}^{N} \alpha_j y_j = 0 \qquad \alpha_j \geq 0, \; j = 1,\ldots,N$$

$$L(\boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j - \frac{1}{2}\sum_{j}^{N}\sum_{k}^{N} \left(\alpha_j \alpha_k y_j y_k \mathbf{z}_j^{\mathrm{T}} \mathbf{z}_k\right)$$

Problema dual

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j - \frac{1}{2} \sum_{j}^{N} \sum_{k}^{N} \left(\alpha_j \alpha_k y_j y_k \mathbf{z}_j^{\mathrm{T}} \mathbf{z}_k \right)$$

$$sa \quad \sum_{j=1}^{N} \alpha_j y_j = 0$$

$$\alpha_j \geq 0, \quad j = 1, \ldots, N$$

$$h(\mathbf{z}) = sign\left(\sum_{j=1}^{N} \alpha_j y_j (\mathbf{z}^T \mathbf{z}_j) + b \right)$$

Kernel trick

$$h(\mathbf{z}) = sign\left(\sum_{j=1}^{N} \alpha_j y_j (\mathbf{z}^{\mathrm{T}} \mathbf{z}_j) + b\right)$$

$$h(\mathbf{x}) = sign\left(\sum_{j=1}^{N} \alpha_j y_j (F(\mathbf{x})^{\mathrm{T}} F(\mathbf{x}_j)) + b\right)$$

$$F(\mathbf{x}_j)^{\mathrm{T}} F(\mathbf{x}_k) = K(\mathbf{x}_j, \mathbf{x}_k) \qquad !$$

$$h(\mathbf{x}) = sign\left(\sum_{j=1}^{N} \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}_k) + b\right)$$

Observação

Este material refere-se às notas de aula do curso EA 072 Inteligência Artificial em Aplicações Industriais da Faculdade de Engenharia Elétrica e de Computação da Unicamp. Não substitui o livro texto, as referências recomendadas e nem as aulas expositivas. Este material não pode ser reproduzido sem autorização prévia dos autores. Quando autorizado, seu uso é exclusivo para atividades de ensino e pesquisa em instituições sem fins lucrativos.