

# Aprendizado Indutivo

## Árvores de Decisão

Jomi F. Hübner

Universidade Federal de Santa Catarina  
Departamento de Automação e Sistemas  
<http://jomi.das.ufsc.br>



# Exemplos de classificação

	Comercial	Distribuída	Internet	Matemática	Tempo Real	Linguagem
1.	S	N	N	S	N	Delphi
2.	S	S	N	S	S	C++

# Exemplos de classificação

	Comercial	Distribuída	Internet	Matemática	Tempo Real	Linguagem
1.	S	N	N	S	N	Delphi
2.	S	S	N	S	S	C++
3.	S	S	S	S	N	Java

# Exemplos de classificação

	Comercial	Distribuída	Internet	Matemática	Tempo Real	Linguagem
1.	S	N	N	S	N	Delphi
2.	S	S	N	S	S	C++
3.	S	S	S	S	N	Java
4.	N	N	S	N	S	Java
5.	N	N	N	S	S	C++
6.	N	S	S	N	N	Java

# Exemplos de classificação

	Tempo	Temperatura	Umidade	Vento	Jogar Tênis
1	sol	quente	alta	fraco	não
2	sol	quente	alta	forte	não
3	nublado	quente	alta	fraco	sim
4	chuva	média	alta	fraco	sim
5	chuva	fria	normal	fraco	sim
6	chuva	fria	normal	forte	não
7	nublado	fria	normal	forte	sim
8	sol	média	alta	fraco	não
9	sol	fria	normal	fraca	sim
10	chuva	média	normal	fraca	sim
11	sol	média	normal	forte	sim
12	nublado	média	alta	forte	sim
13	nublado	quente	normal	fraco	sim
14	chuva	média	alta	forte	não

# ID3

1 **function** id3( $\mathcal{S}$ ,  $\mathcal{A}$ ,  $t$ ): Tree

**Input:** um conjunto de exemplos,  $\mathcal{S}$ ; um conjunto de atributos  $\mathcal{A}$ ;  
um atributo alvo  $t$

**Output:** uma árvore de decisão

2  $n \leftarrow$  new nodo ;

3 **if** todos os exemplos  $s \in \mathcal{S}$  tem a mesma decisão  $d$  **then**

4 |  $n$  é uma folha com a decisão  $d$  ;

5 **else**

6 |  $b \leftarrow$  é melhor atributo em  $\mathcal{A}$  para classificar  $\mathcal{S}$  ;

7 | **foreach**  $v \in \text{Values}(b)$  **do**

8 | |  $S_v \leftarrow$  subconjunto de  $\mathcal{S}$  onde  $b = v$  ;

9 | | adiciona um ramo ao nodo  $n$  ;

10 | | liga este ramo ao resultado de id3( $S_v$ ,  $\mathcal{A} \setminus \{b\}$ ,  $t$ ) ;

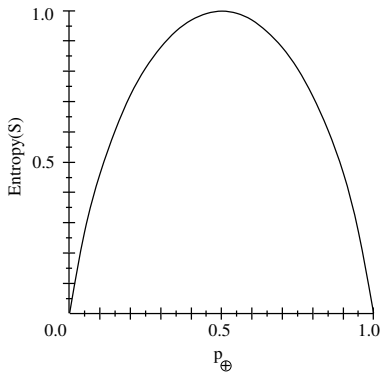
11 **return**  $n$

# Escolha do melhor atributo – entropia

$$\text{Entropia}(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

onde

- $S$  é um conjunto de exemplos
- $c$  é o número de decisões
- $p_i$  é a proporção de exemplos em  $S$  com decisão  $i$



## Escolha do melhor atributo – **ganho**

$$\text{Ganho}(S, A) = \text{Entropia}(S) - \sum_{v \in A^v} \frac{|S_v|}{|S|} \text{Entropia}(S_v)$$

onde

- $S$  é um conjunto de exemplos
- $A$  é um atributo
- $A^v$  é o conjunto de valores do atributo  $A$
- $S_v$  é subconjunto de  $S$  onde  $A = v$



# Exemplo

$S$	$Entropia(S)$
{ 1, 2, 3 } (Comercial = sim)	1.58
{ 4, 5, 6 } (Comercial = não)	0.92
{ 3, 4, 6 } (Internet = sim)	0
{ 1, 2, 5 } (Internet = não)	0.92

## Atributos com domínio contínuo

Como classificar se, por exemplo, a temperatura fosse dada em graus (e não “quente”, “frio”, ...)?

Solução: criar um atributo  $A_c$  que é verdadeiro se  $A < c$

Temperatura:	10	15	20	22	25	28	35
Decisão:	N	N	S	S	S	S	N

Candidatos:

- $A_{17,5}$  (de  $(15 + 20)/2$ )
- $A_{31,5}$  (de  $(28 + 35)/2$ )

Calcula-se o ganho de cada um para seleccionar o melhor

## Atributos com domínio contínuo

Como classificar se, por exemplo, a temperatura fosse dada em graus (e não “quente”, “frio”, ...)?

Solução: criar um atributo  $A_c$  que é verdadeiro se  $A < c$

Temperatura:	10	15	20	22	25	28	35
Decisão:	N	N	S	S	S	S	N

Candidatos:

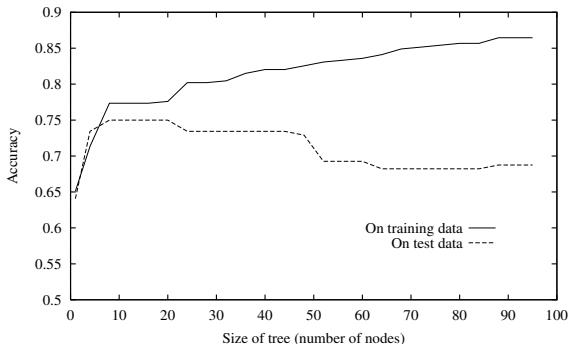
- $A_{17,5}$  (de  $(15 + 20)/2$ )
- $A_{31,5}$  (de  $(28 + 35)/2$ )

Calcula-se o ganho de cada um para seleccionar o melhor

# Problema: **decorar** exemplos

## Definition (overfitting)

Uma hipótese de classificador *decora* os exemplos de treinamento quanto (i) uma outra hipótese classifica pior esses exemplos, **mas** (ii) classifica melhor um conjunto de teste



# Problema: **decorar** exemplos

Razões:

- erros nos exemplos
- poucos exemplos

Soluções:

- impedir o crescimento da árvore
- deixar a árvore “crescer”, mas depois podar
  - para cada nodo, se sua remoção não tornar a árvore pior no conjunto de teste, então remover o nodo.



# Características do ID3

- Escolhe a hipótese mais geral numa busca estilo subida da montanha  
(sem back-tracking, sujeito a máximos locais)
- O ganho é a heurística
- Tesoura de *Occam*: preferir o mais simples  
↔ a menor árvore consistente
- Pode ser facilmente alterado para considerar valores ausentes ou incorretos (C4.5)

# Quando considerar árvores de decisão

- Os exemplos são descritos por pares atributo-valor
- A função de classificação tem imagem discreta
- Hipóteses disjuntivas podem ser necessárias para classificação
- Dados com erros

# Material bibliográfico

-  Mitchell, T. M. (1997).  
*Machine Learning*. Chaper 3.  
McGraw-Hill.
-  Russell, S. J. and Norvig, P. (2010).  
*Artificial Intelligence: A Modern Approach*. Chapter 18.  
Prentice Hall, New Jersey, 3 edition.