Inteligência Artificial

Profa. Patrícia R. Oliveira
EACH / USP

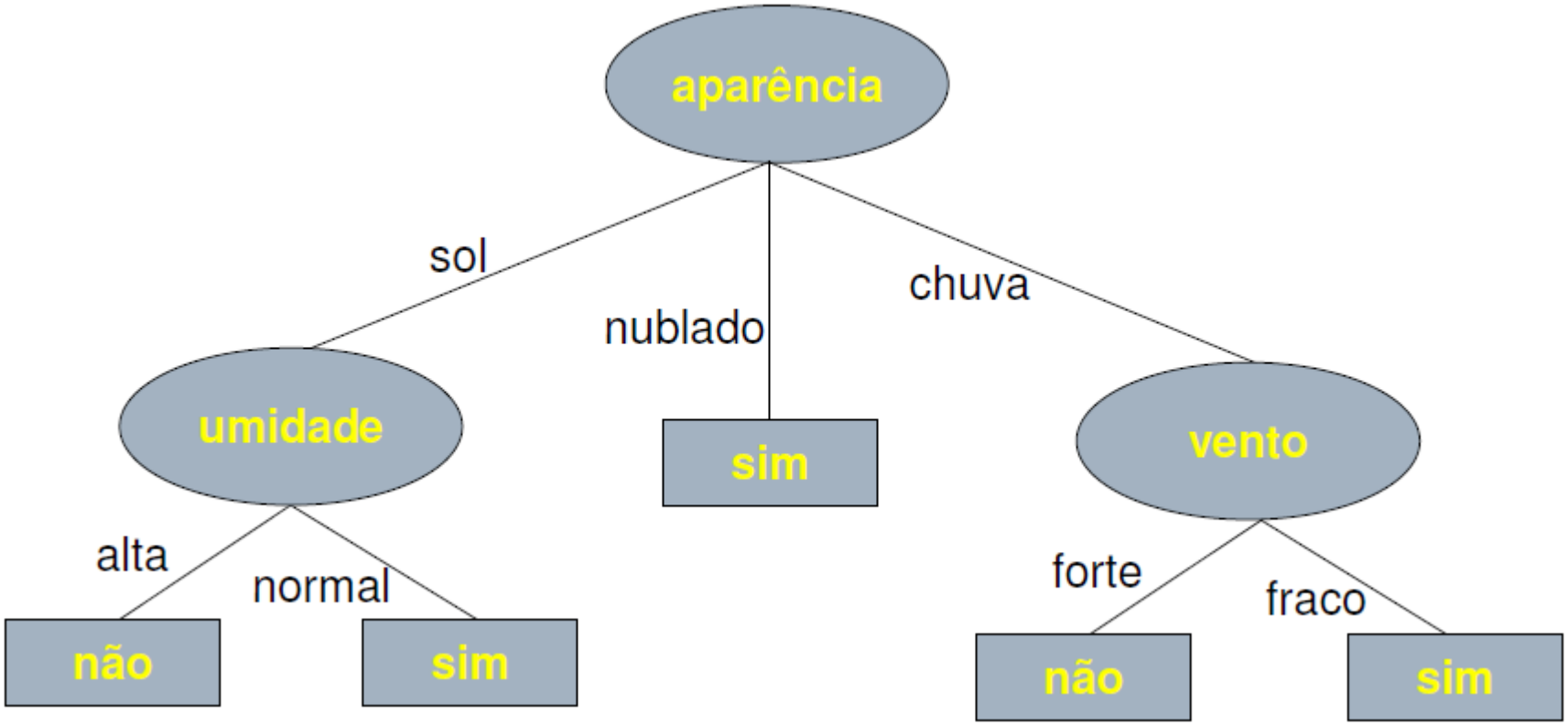
Parte 3 – Aprendizado Simbólico: Árvores de Decisão

Este material é parcialmente baseado em slides
do Prof. Thiago Pardo (ICMC/USP)

Árvores de Decisão (ADs)

- Trata-se de um modelo prático de uma função recursiva que determina o valor de uma variável (realização de um teste) e, baseando-se neste valor, executa-se uma ação.
- Esta ação pode ser a escolha de outra variável (realização de outro teste) ou a saída (classe).

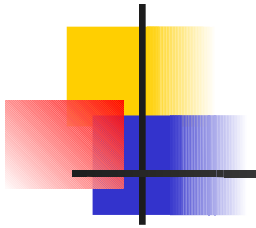
Exemplo de árvore de decisão



Árvores de Decisão (ADs)

- As árvores de decisão são treinadas de acordo com um conjunto de exemplos previamente classificados
  - Aprendizado supervisionado
- Posteriormente, outros exemplos fora do conjunto de treinamento devem ser classificados de acordo com essa mesma árvore.
- É possível ter uma visão gráfica da tomada de decisão.

Quando usar ADs?

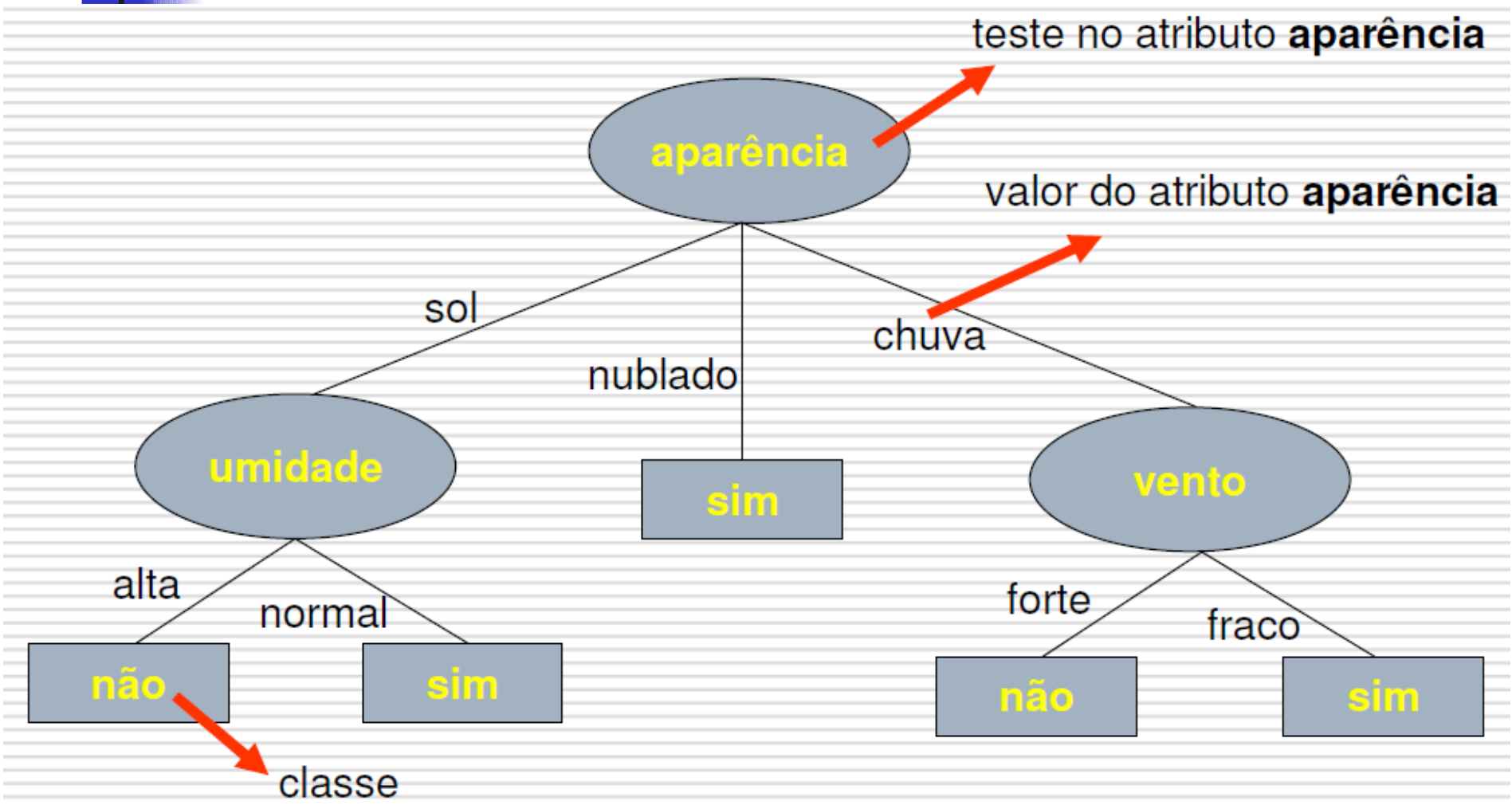


- Instâncias (exemplos) são representadas por pares atributo-valor.
- Função objetivo assume apenas valores discretos.
- Hipóteses disjuntivas podem ser necessárias.
- Conjunto de treinamento possivelmente corrompido por ruído (valores errados, incompletos ou inconsistentes).
- Exemplos:
  - Diagnóstico médico, diagnóstico de equipamentos, análise de crédito.

Estrutura

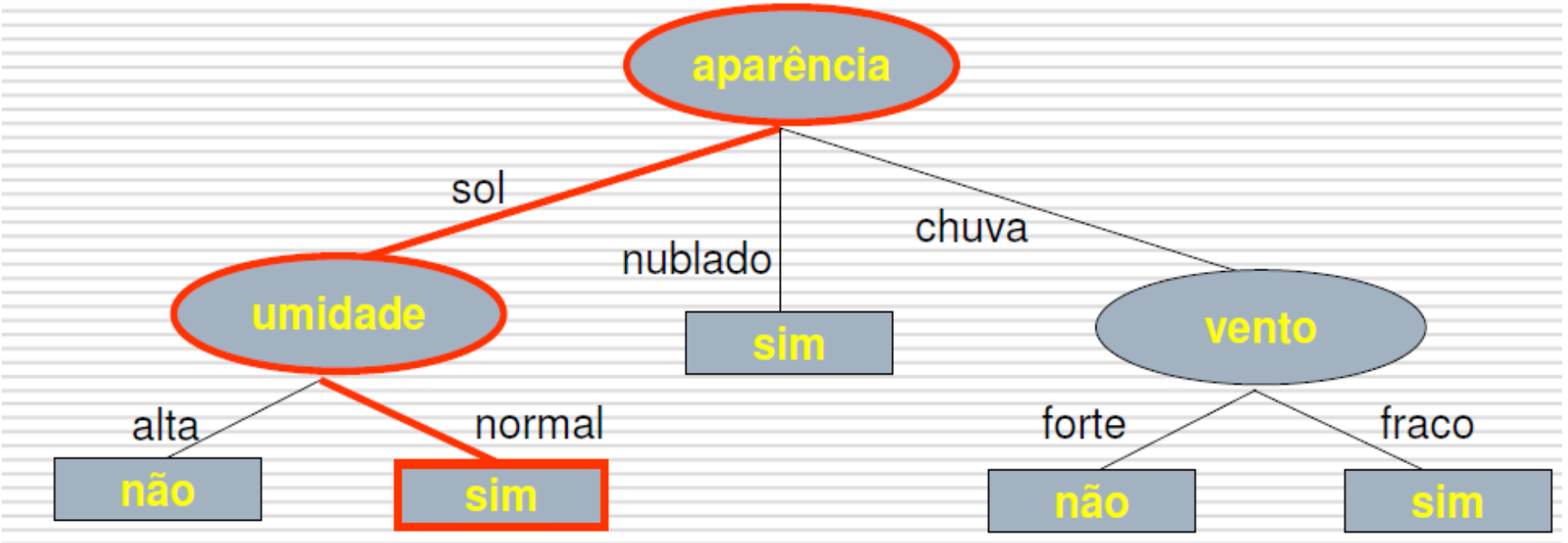
- Uma AD alcança sua decisão executando uma sequência de testes.
- Cada nó interno na árvore corresponde a um teste do valor de um dos atributos do conjunto de exemplos T.
- As ramificações a partir de um nó interno são rotuladas com os possíveis valores do teste realizado nesse nó.
- Cada nó-folha na árvore especifica a classe a ser retornada se aquela folha for alcançada.

Exemplo


teste no atributo **aparência**
aparência
valor do atributo **aparência**
sol
nublado
chuva
umidade
sim
vento
alta
normal
forte
fraco
não
sim
não
sim
classe

Exemplo

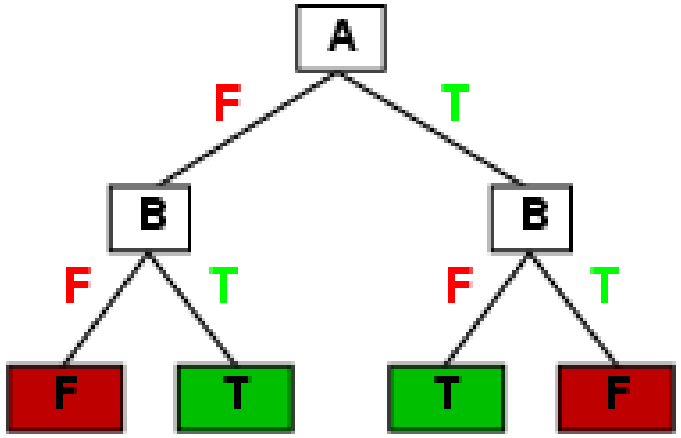
- Para classificar um novo exemplo, basta começar pela raiz, seguindo cada teste até que a folha seja alcançada.



AD para funções booleanas

- Qualquer função booleana pode ser escrita como uma árvore de decisão.
- Cada linha na tabela verdade corresponde a um caminho na árvore.

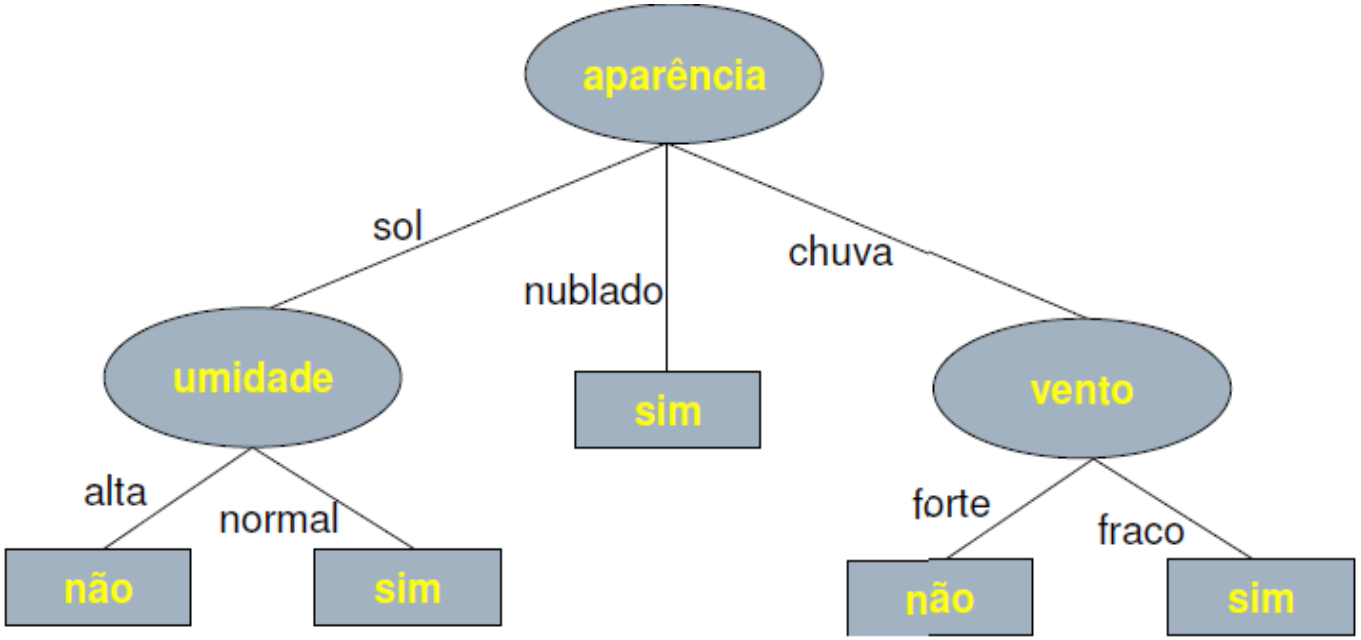
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Exercício

- Determine as tabelas verdade e as correspondentes ADs para as seguintes funções booleanas:
  - A not B
  - A ∨ [B ∧ C]

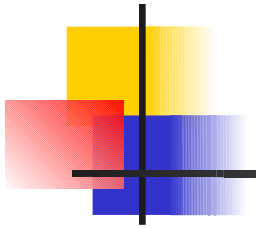
Transformação de uma AD em regras



- IF (aparência = sol) ∧ (umidade = alta) THEN JogaTenis = não
- IF (aparência = sol) ∧ (umidade = normal) THEN JogaTenis = sim

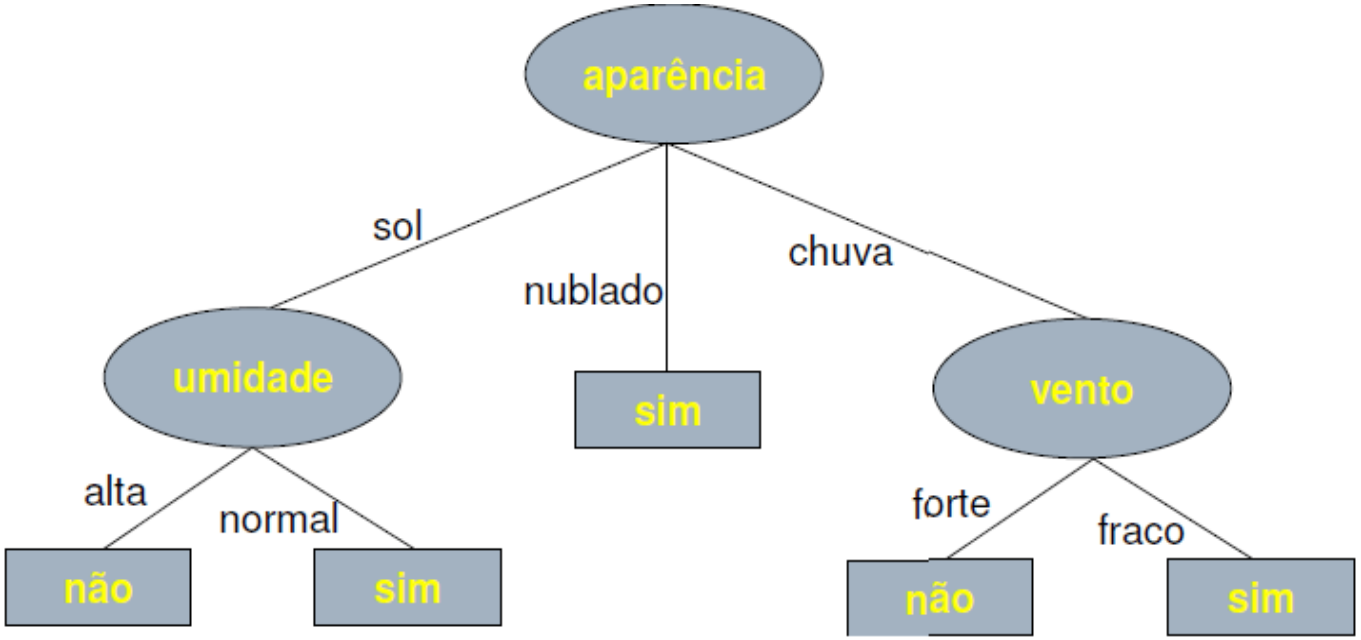
.....

Transformação de uma AD em regras



- Toda AD pode ser representada como um conjunto de regras de classificação, sendo que cada regra é gerada percorrendo-se um caminho na AD, da raiz até uma das folhas.
- A regra prediz a classe associada à folha.
- A parte condicional da regra é obtida pela conjunção das condições de cada nó no caminho.
- Cada conceito (classe) induzido pela AD pode ser expresso por uma disjunção de conjunções.
  - Regra para o conceito JogarTenis?

Transformação de uma AD em regras



- Conceito JogarTennis:

((aparência = sol) ∧ (umidade = normal)) v (aparência = nublado) v ((aparência = chuva) ∧ (vento = fraco))

Construção de uma AD

- A idéia base:

1. Escolher um atributo.
2. Estender a árvore adicionando um ramo para cada valor do atributo.
3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido).
4. Para cada folha
   - 4.1. Se todos os exemplos são da mesma classe, associar essa classe à folha.
   - 4.2. Senão repetir os passos 1 a 4.

Atributo de particionamento

- A chave para o sucesso de um algoritmo de aprendizado de AD depende do critério utilizado para escolher o atributo que particiona (atributo de teste) o conjunto de exemplos em cada iteração
- Questão: como escolher o atributo de particionamento?

Outro Exemplo

- Decidir se devo esperar por uma mesa em um restaurante, dados os atributos:

1. **Alternate:** há um restaurante alternativo na redondeza?
2. **Bar:** existe um bar confortável onde eu possa esperar?
3. **Fri/Sat**: hoje é sexta ou sábado?
4. **Hungry:** estou com fome?
5. **Patrons:** número de pessoas no restaurante (None, Some, Full).
6. **Price:** faixa de preços ($, $$, $$$)
7. **Raining:** está chovendo?
8. **Reservation:** tenho reserva?
9. **Type:** tipo do restaurante (French, Italian, Thai, Burger)
10. **WaitEstimate:** tempo de espera estimado (0-10, 10-30, 30-60, >60).

Outro Exemplo (cont.)

- Exemplos disponíveis:

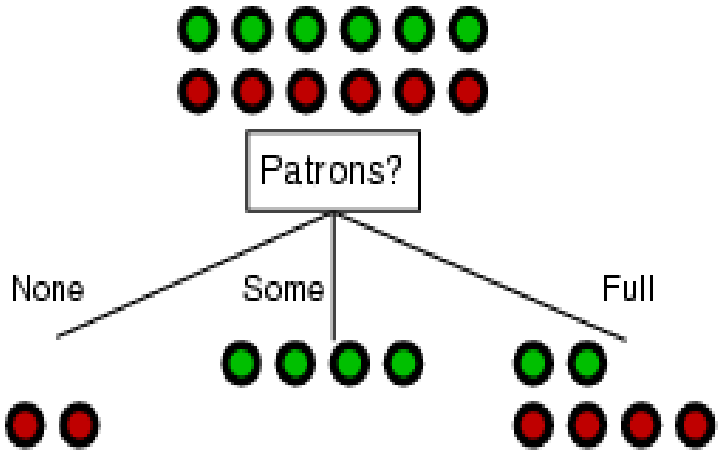
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

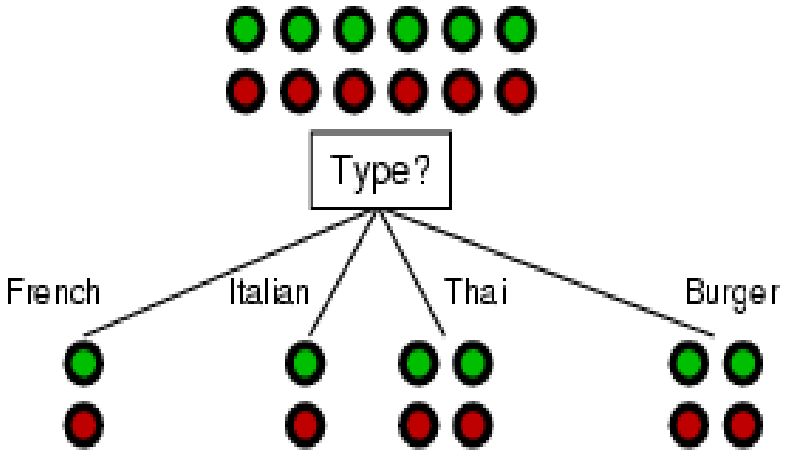
- Classificação dos exemplos em positivo (T) ou negativo (F).

Escolha de Atributos

- Objetivo: encontrar a menor árvore que seja consistente com os exemplos.
- Ideia: (recursivamente) encontre o atributo "mais significante" como raiz da sub-árvore.
- Um bom atributo divide os exemplos em subconjuntos que (preferivelmente) são "todos positivos" ou "todos negativos".

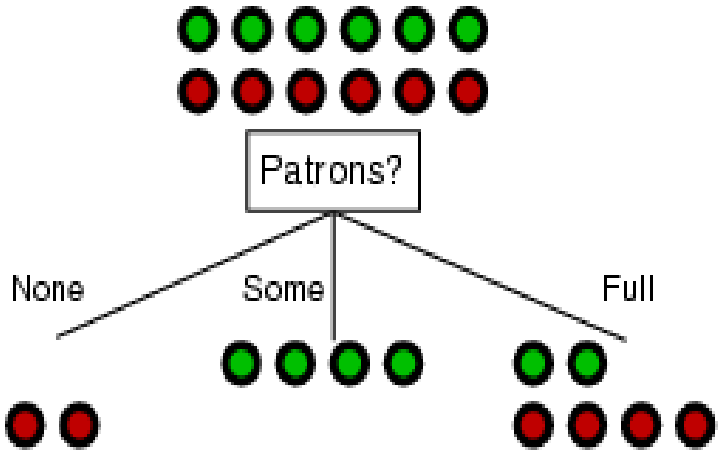
Escolha de Atributos

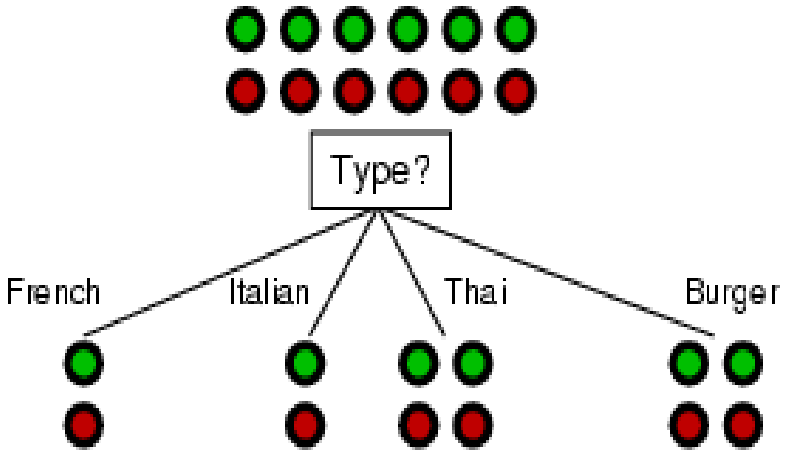




- Pergunta: qual dos dois atributos deve ser escolhido como raiz da AD?

Escolha de Atributos





- Resposta: *Patrons* é um atributo melhor do que *Type* para ser raiz da AD.

Algoritmo ID3 (informal)

- Se existem alguns exemplos positivos e alguns negativos, escolha o melhor atributo para dividi-los.
- Se todos os exemplos restantes forem positivos (ou todos negativos), então terminamos: podemos responder **Sim** ou **Não**.
- Se não resta nenhum exemplo, nenhum exemplo desse tipo foi observado. Então retorna-se um valor-padrão calculado a partir da classificação de maioria no pai do nó.

Algoritmo ID3 (informal)

- Se não resta nenhum atributo, mas há exemplos positivos e negativos, quer dizer que esses exemplos tem exatamente a mesma descrição, mas classificações diferentes.
- Isso acontece quando:
  - alguns dados estão incorretos; dizemos que existe ruído nos dados;
  - os atributos não fornecem informações suficientes para descrever a situação completamente;
  - o domínio não é completamente determinístico;
  - saída simples: utilizar uma votação pela maioria.

Algoritmo ID3 (formal)

- Algoritmo recursivo
- ID3(Examples, Target_attribute, Attributes)
  - Examples: exemplos de treinamento
  - Target_attribute: atributo cujos valores devem ser previstos pela AD
  - Attributes: atributos que podem ser testados pela AD.

```
ID3 (Examples, Target_attribute, Attributes)

  Crie um nó raiz Root para a AD
  Se todos os exemplos forem positivos, retorne o nó Root, com rótulo +
  Se todos os exemplos forem negativos, retorne o nó Root, com rótulo -
  Se Attributes for vazio, retorne o nó Root, com rótulo = valor mais comum
     de Target_Attribute em Examples
  Caso Contrário Begin
    A← o atributo que melhor classifica Examples
    O atributo para Root ← A
    Para cada possível valor, v_i para A
      Adicione um novo ramo abaixo de Root correspondente ao teste A= v_i
      Seja Examples_vi o subconjunto de Examples com valor v_i para A
      Se Examples_vi for vazio Então
          Abaixo desse novo ramo, adicione um novo nó folha, com rótulo =
          valor mais comum de Target_Attribute em Examples
      Senão
          Abaixo desse novo ramo, adicione a subárvore
            ID3 (Examples_vi, Target_attribute, Attributes - {A})
  End
  Return Root
```

Como definir o que é um Atributo melhor?

- A escolha de atributos deve minimizar a profundidade da árvore de decisão;
- Escolher um atributo que vá o mais longe possível na classificação exata de exemplos;
- Um atributo perfeito divide os exemplos em conjuntos que são todos positivos ou todos negativos.
- Solução: medir os atributos a partir da quantidade esperada de informações fornecidas por ele.

Como definir o que é um Atributo melhor?

- É necessária uma boa medida quantitativa.
- O algoritmo ID3 utiliza uma medida denominada de ganho de informação
  - propriedade estatística.
  - mede o quão bem um atributo separa os exemplos de treinamento de acordo com a meta de classificação.
  - baseada na medida de entropia.

Entropia

- Caracteriza a heterogeneidade de classes em uma coleção de exemplos.
- Seja uma coleção S, contendo exemplos positivos e negativos de um determinado conceito, a entropia de S é dada por:

$$Entropy(S) \equiv - p_{\oplus} \log_2 p_{\oplus} - p_{-} \log_2 p_{-}$$

em que:

$p_{\oplus}$: proporção de exemplos positivos em S.

p_{-} : proporção de exemplos negativos em S.

Obs: para o cálculo de entropia, define-se $0 \log_2 0 = 0$.

Entropia – Exemplo

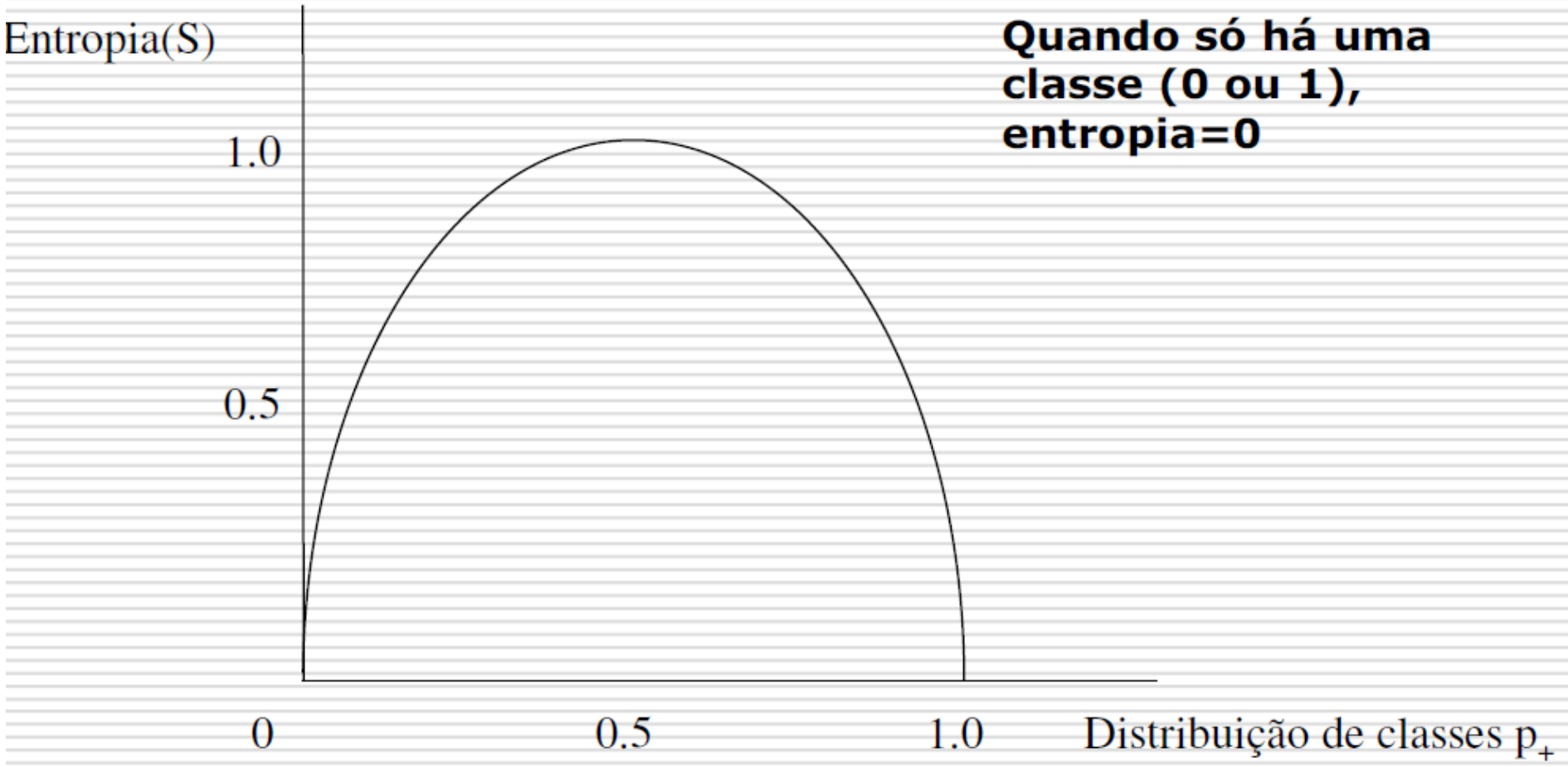
- Suponha que S é uma coleção de 14 exemplos, sendo 9 desses positivos e 5 negativos.
- Notação: [9+, 5–]
- Entropia de S:

$$Entropy([9+,5-]) =$$
$$-(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

Entropia

- Note que a entropia é igual a 0 se todos os membros de S pertencerem a mesma classe (os dados estão perfeitamente classificados).
- A entropia será igual a 1 se a coleção apresentar um número igual de exemplos positivos e negativos.
- No caso da classificação booleana, a medida da entropia varia de 0 ("perfeitamente classificado") a 1 ("totalmente aleatório").

Entropia


Entropia(S)
1.0
0.5
0
0.5
1.0
Distribuição de classes p_+
Quando só há uma classe (0 ou 1), entropia=0

Entropia – Caso Geral

- Qualquer que seja o número de classes em um conjunto de dados S, a entropia de S é dada pela fórmula:

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

em que p_i é a proporção de exemplos da classe i pertencentes a S e c é o número total de classes.

Ganho de Informação

- Ganho(S,A): está relacionado à redução esperada na entropia do conjunto S pelo particionamento deste, de acordo com os valores observados do atributo A.

$$Ganho(S,A) \equiv Entropia(S) - \sum_{v \in Valores(A)} \frac{|Sv|}{|S|} Entropia(Sv)$$

em que:

Valores(A): conjunto de todos os valores possíveis para A.

S_v: subconjunto de S para o qual A tem o valor v.

Exemplo

- Atributo *Vento*, com valores *forte* e *fraco.*
- Coleção S com 14 exemplos, sendo que 9 são positivos e 5 são negativos ([9+,5-]).
- Desses 14 exemplos, suponha que 6 dos exemplos positivos e dois exemplos negativos tem *Vento* = *fraco* [6+,2-] e o resto tem *Vento* = *forte* ([3+,3-]).
- Portanto
  - Valores(*Vento*) = *fraco*, *forte*
  - Distribuição de S = [9+,5-]
  - Distribuição de S_{fraco} = [6+,2-]
  - Distribuição de S_{forte} = [3+,3-]

Exemplo

$$Ganho(S, Vento) \equiv Entropia(S) - \sum_{v \in \{fraco, forte\}} \frac{|Sv|}{|S|} Entropia(Sv)$$

$$Ganho(S, Vento) \equiv Entropia(S) - \left(\frac{8}{14}\right) Entropia(Sfraco) - \left(\frac{6}{14}\right) Entropia(Sforte)$$

$$Ganho(S, Vento) \equiv 0.940 - \left(\frac{8}{14}\right) 0.811 - \left(\frac{6}{14}\right) 1.00$$

$$Ganho(S, Vento) \equiv 0.048$$

Exercício

- Qual é a entropia desse conjunto de treinamento?
- Qual o ganho de informação do atributo A2 em relação a esse conjunto de treinamento?

Instância	A1	A2	Classe
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

Construindo uma AD

dia	aparência	temperatura	umidade	vento	jogar tênis
D1	ensolarado	quente	alta	fraco	*não*
D2	ensolarado	quente	alta	forte	*não*
D3	nublado	quente	alta	fraco	*sim*
D4	chuva	moderada	alta	fraco	*sim*
D5	chuva	fria	normal	fraco	*sim*
D6	chuva	fria	normal	forte	*não*
D7	nublado	fria	normal	forte	*sim*
D8	ensolarado	moderada	alta	fraco	*não*
D9	ensolarado	fria	normal	fraco	*sim*
D10	chuva	moderada	normal	fraco	*sim*
D11	ensolarado	moderada	normal	forte	*sim*
D12	nublado	moderada	alta	forte	*sim*
D13	nublado	quente	normal	fraco	*sim*
D14	chuva	moderada	alta	forte	*não*

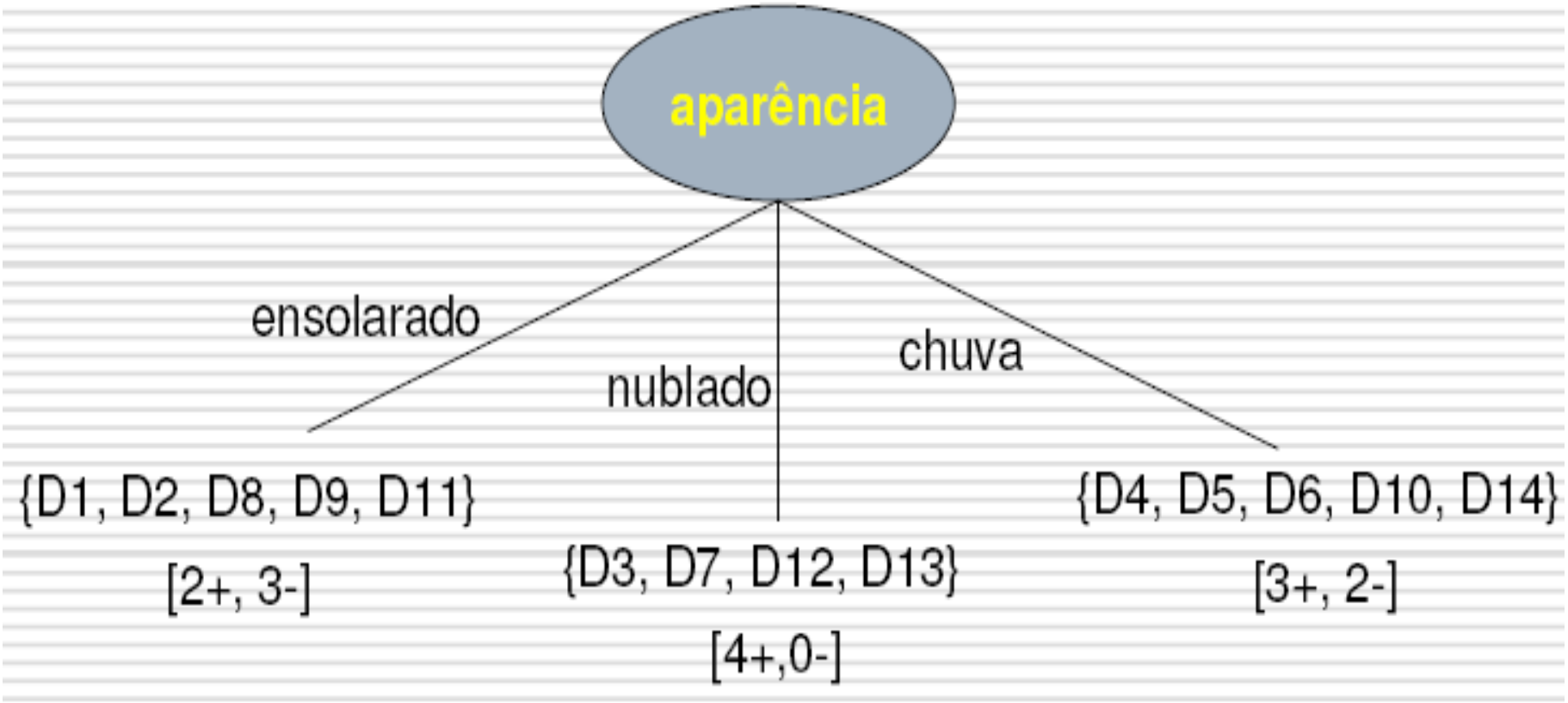
Construindo uma AD

- Qual deverá ser o nó raiz da árvore?
  - O ganho de informação deve ser calculado para cada atributo do conjunto de treinamento (menos o atributo classe).
  - O atributo que resultar no maior ganho de informação é selecionado como atributo de teste.

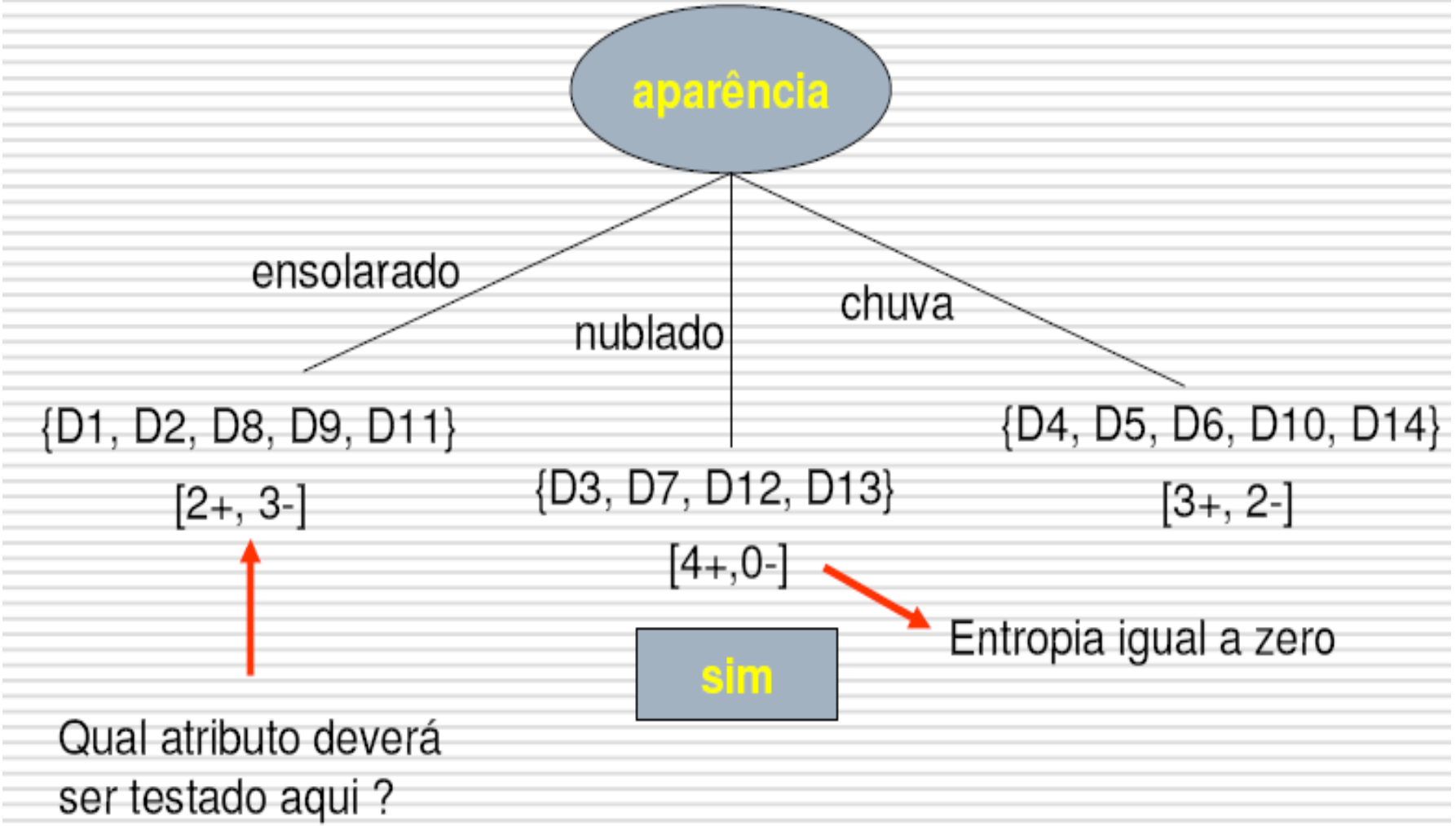
Construindo uma AD

- Para cada atributo:
  - Ganho(S, aparência) = 0.246
  - Ganho(S, umidade) = 0.151
  - Ganho(S, vento) = 0.048
  - Ganho(S, temperatura) = 0.029
- De acordo com a medida de ganho de informação, o atributo **aparência** é o que melhor prediz o atributo classe, **jogar_tênis**, sobre o conjunto de treinamento.

Construindo uma AD



Construindo uma AD



Construindo uma AD

- Todo exemplo para o qual **aparência=nublado** tem o atributo classe **jogar_tênis=sim**
  - é criado um nó folha com a classificação **jogar_tênis=sim** (chamada recursiva).
- Os nós descendentes correspondentes a **aparência=ensolarado** e **aparência=chuva** ainda tem entropias diferentes de zero
  - deverá ser criada uma nova AD abaixo de cada um desses ramos (chamada recursiva).

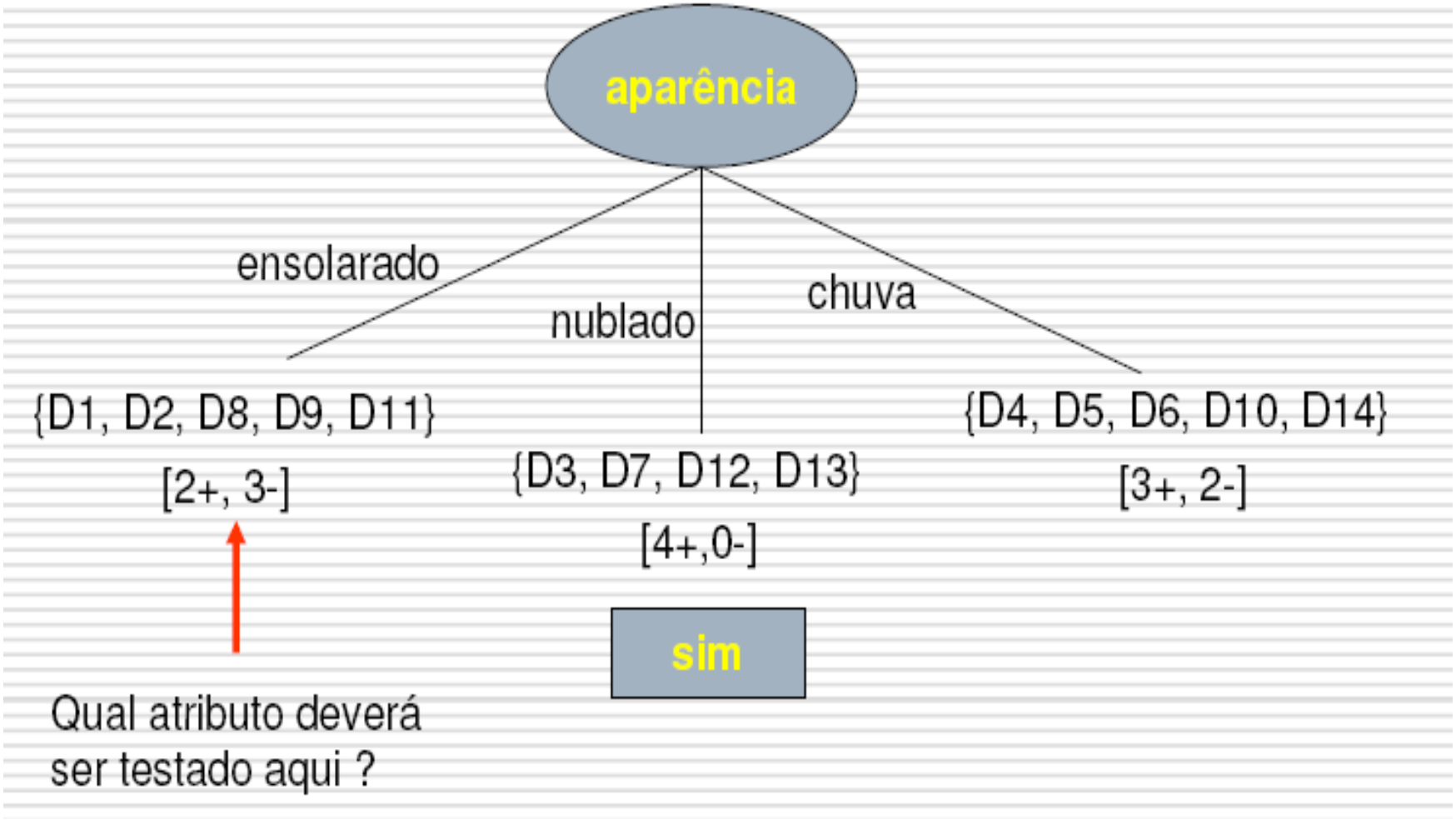
Construindo uma AD

- O processo de selecionar um novo atributo e particionar os exemplos de treinamento é repetido para cada nó descendente não folha.
  - usa-se somente os exemplos associados àquele nó.
  - atributos que já foram usados nos nós ascendentes deste são excluídos do processo de seleção.
    - um atributo pode aparecer no máximo uma vez ao longo de um mesmo caminho na AD.

Construindo uma AD

- O processo de selecionar e particionar continua até que uma dessas condições seja atingida:
  - (1) todos os atributos já foram usados ao longo do caminho na AD;
  - (2) os exemplos de treinamento associados ao nó folha possuem o mesmo valor para o atributo classe.
    - a entropia desse nó é igual a zero.

Construindo uma AD



Construindo uma AD

dia	aparência	temperatura	umidade	vento	jogar tênis
D1	ensolarado	quente	alta	fraco	*não*
D2	ensolarado	quente	alta	forte	*não*
D3	nublado	quente	alta	fraco	*sim*
D4	chuva	moderada	alta	fraco	*sim*
D5	chuva	fria	normal	fraco	*sim*
D6	chuva	fria	normal	forte	*não*
D7	nublado	fria	normal	forte	*sim*
D8	ensolarado	moderada	alta	fraco	*não*
D9	ensolarado	fria	normal	fraco	*sim*
D10	chuva	moderada	normal	fraco	*sim*
D11	ensolarado	moderada	normal	forte	*sim*
D12	nublado	moderada	alta	forte	*sim*

Construindo uma AD

- $S_{ensolarado}$ = {D1, D2, D8, D9, D11}

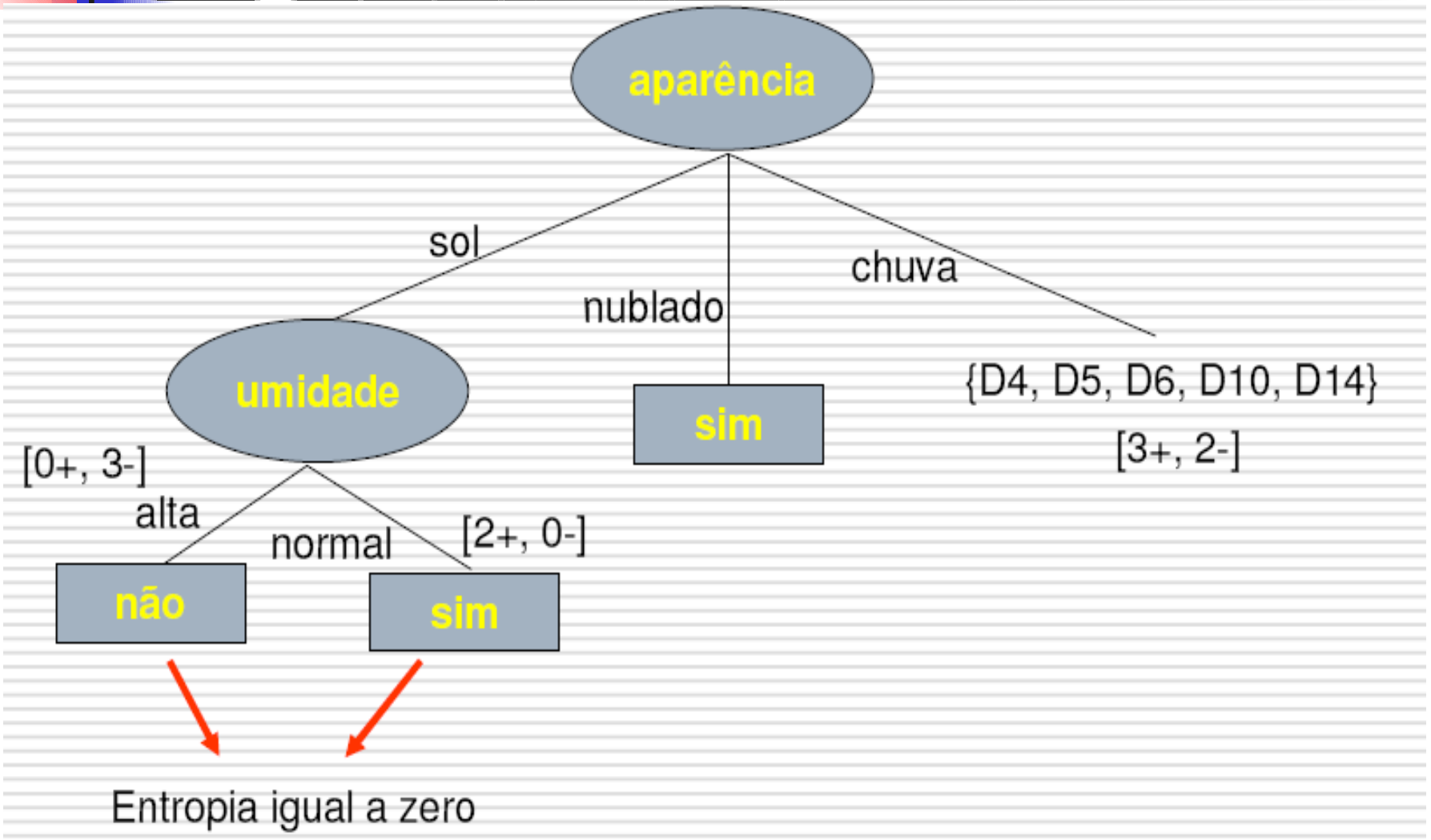
$$Ganho(S_{ensolarado}, umidade) = 0.970-(3/5)0.0 - (2/5)0.0 = 0.970$$

$$Ganho(S_{ensolarado}, temperatura) = 0.970-(2/5)1.0 - (2/5)0.0 - (1/5)0.0 = 0.570$$

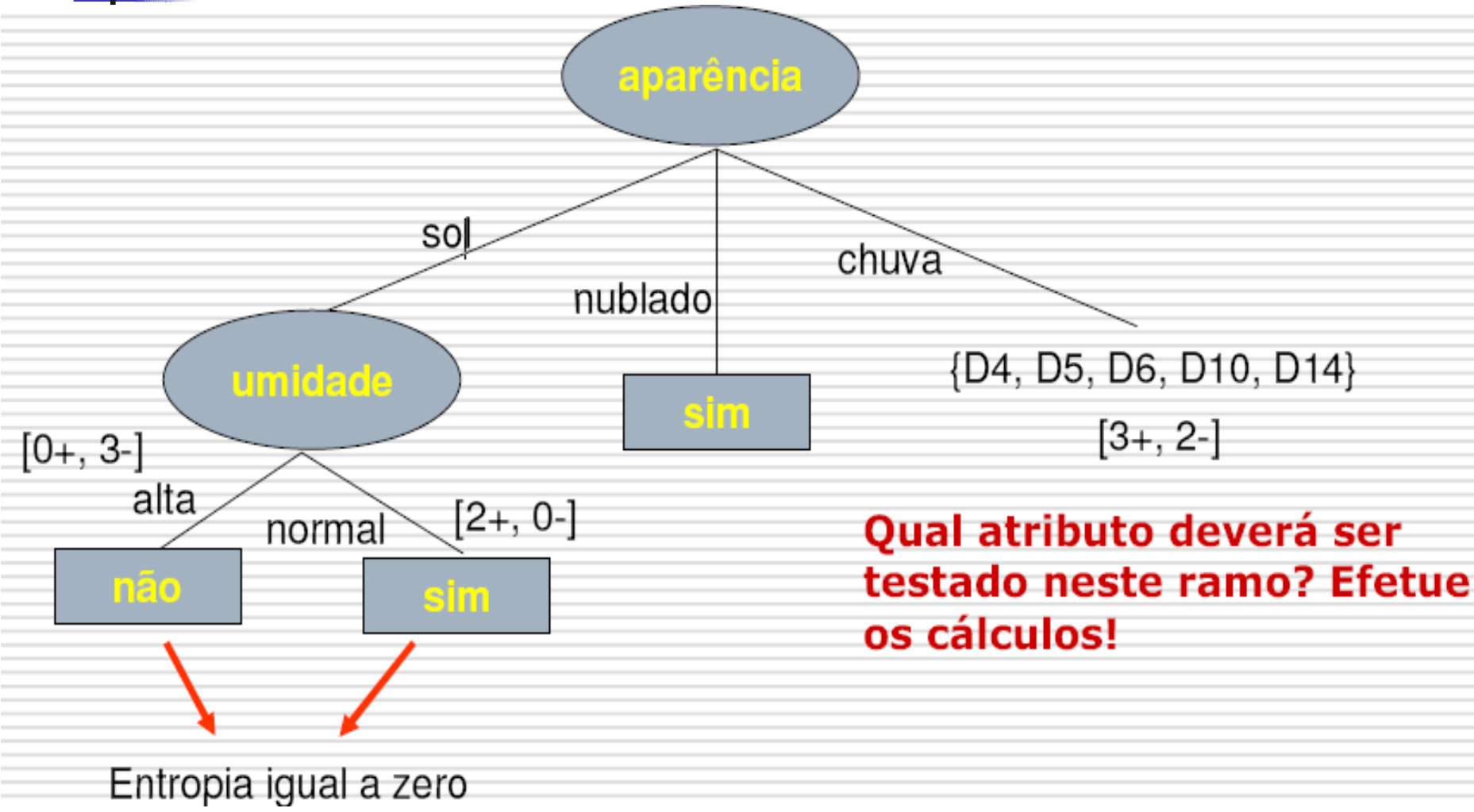
$$Ganho(S_{ensolarado}, vento) = 0.970-(2/5)1.0 - (3/5)0.918 = 0.019$$

- Nesse caso, o maior ganho de informação está no atributo **umidade**.

Construindo uma AD

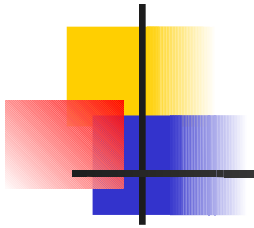


Exercício



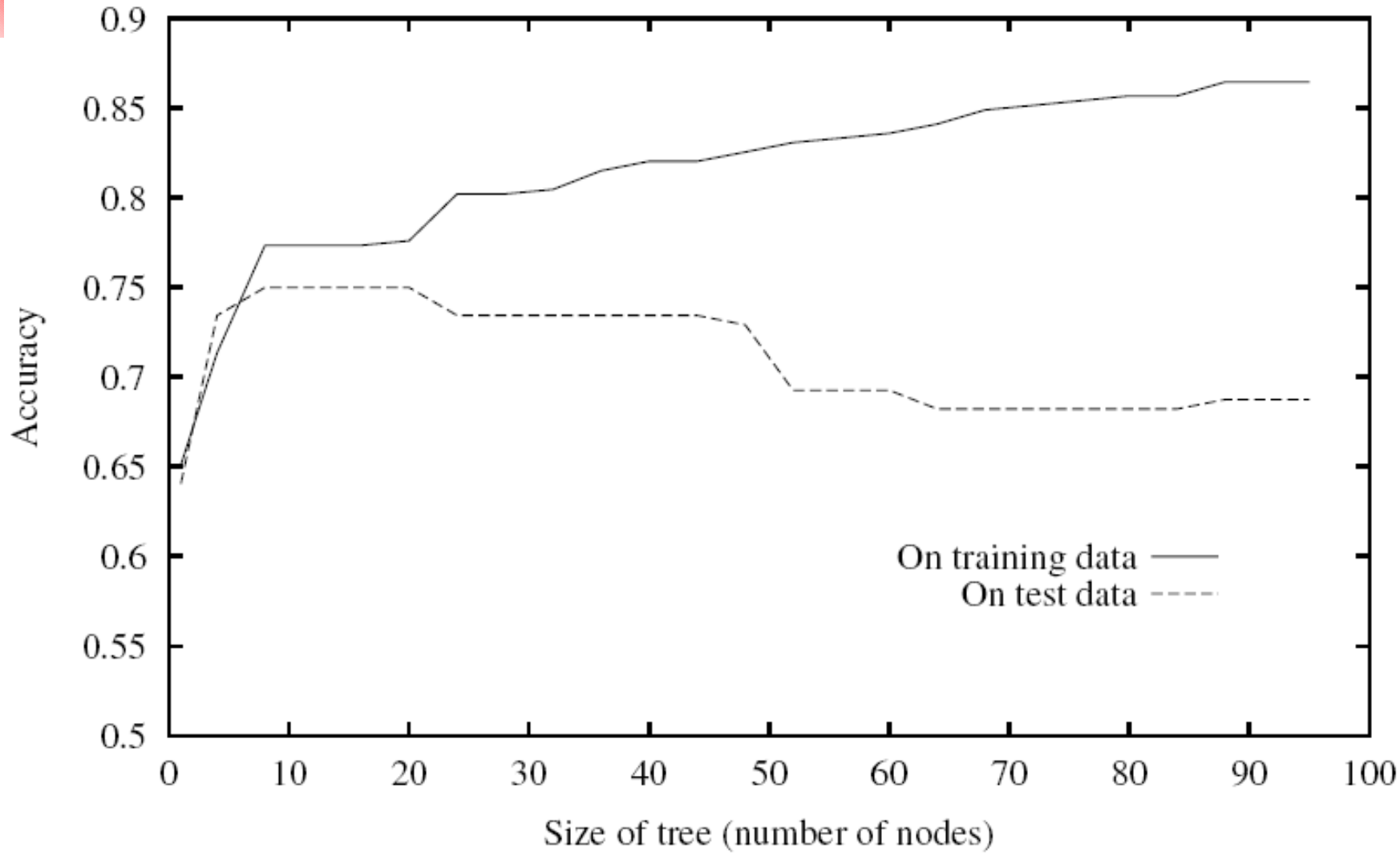
Qual atributo deverá ser testado neste ramo? Efetue os cálculos!

Overfitting em ADs



- O algoritmo ID3 expande cada ramificação na AD até que todos os exemplos estejam classificados corretamente.
  - Indesejável quando há ruído nos dados.
  - Pode levar a árvores superajustadas aos exemplos de treinamento (overfitting).

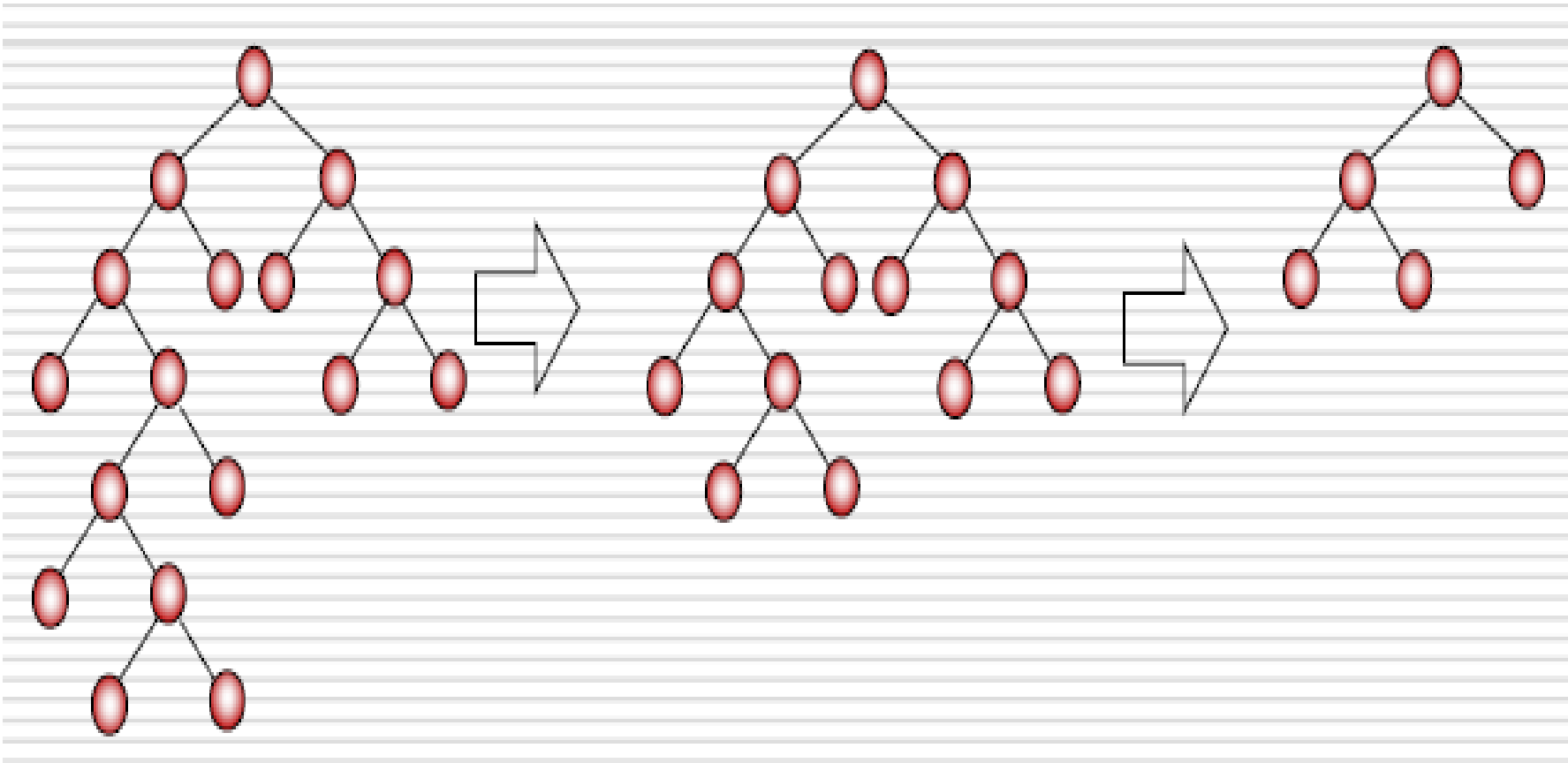
Overfitting em ADs



Evitando Overfitting em ADs

- Duas abordagens básicas:
  - parar de expandir a AD quando um novo particionamento não for significativo (pré-poda).
    - difícil de estimar precisamente quando parar.
  - construir a AD completa e depois realizar um procedimento de poda (pós-poda).

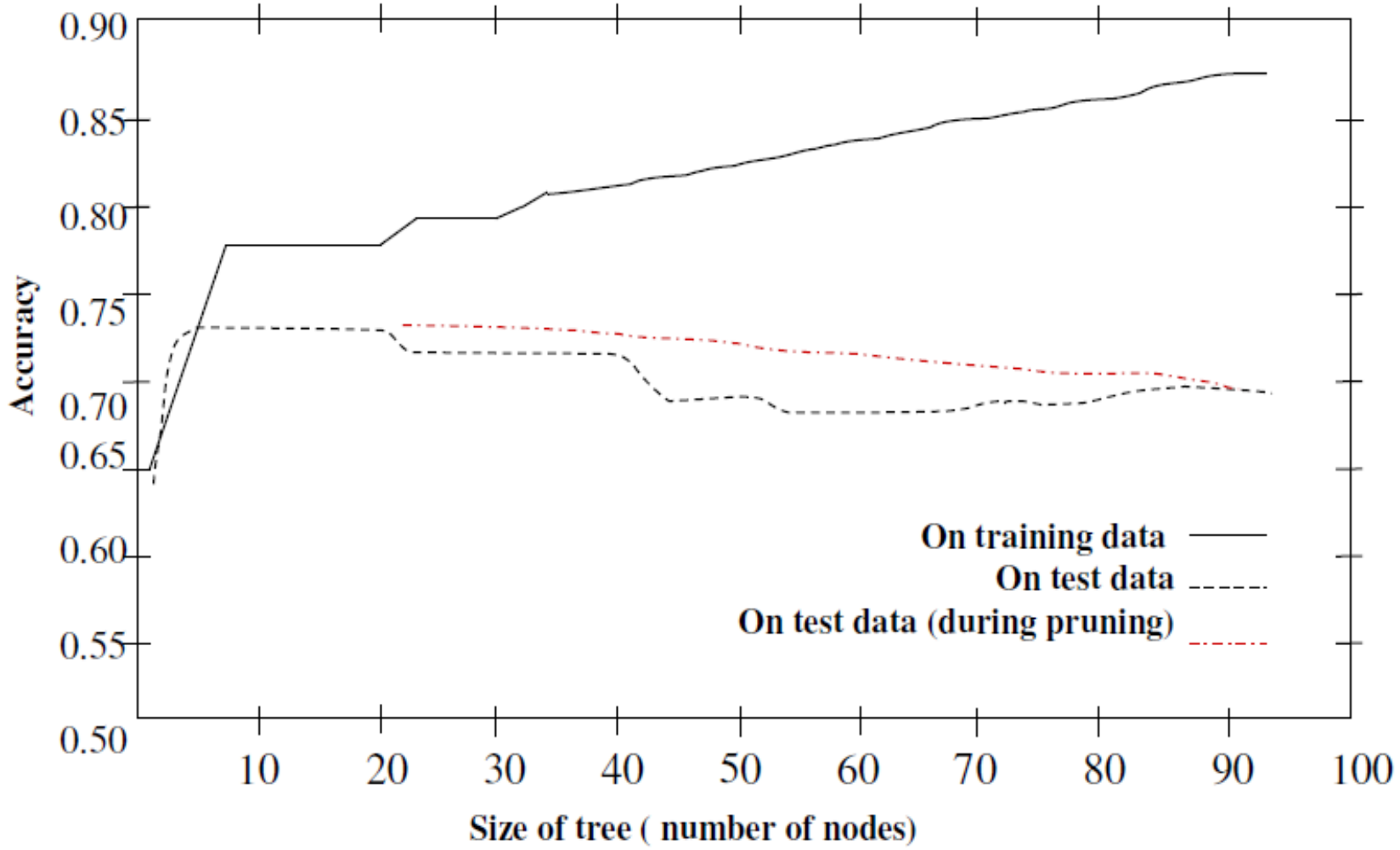
Exemplo de Poda



Poda por Redução de Erro

- Divide o conjunto de dados em conjunto de treinamento, conjunto de validação e conjunto de teste.
- Podar um nó significa remover a subárvore a partir daquele nó, tornando-o um nó-folha e designando-lhe a classificação mais comum.
- Faça até que seja prejudicial (aumente o erro do modelo):
  - 1) avalie o impacto da poda sobre o conjunto de validação para cada nó da AD.
  - 2) remova o nó que diminua mais significativamente o erro do modelo.

Efeito da Poda por Redução de Erro



Pós-poda de Regras

- 1) Converta a AD em um conjunto de regras equivalente.
- 2) Pode cada regra de forma independente, removendo pré-condições que resultem na melhora da precisão da AD.
- 3) Ordene as regras finais de acordo com as suas precisões estimadas.
  - considere essa sequência nas classificações subsequentes.

Atributos Contínuos

- O que fazer com atributos contínuos, que não são discretos?
- Solução:
  - Valores dos atributos são discretizados, ou seja, divididos em intervalos.
  - Novos atributos booleanos são criados e testados em função dos intervados definidos.

Atributos Contínuos: ID3

- Exemplo:

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

- Escolhe-se o valor que melhor divide o espaço, de forma que haja o maior ganho de informação.
- Foi mostrado que esse valor fica entre os limites das mudanças de classes (entre 48 e 60; entre 80 e 90).

$$\begin{cases} (48+60)/2 & \text{Temperature}>54 \\ (80+90)/2 & \text{Temperature}>85 \end{cases}$$

- Temperature>54 causa maior ganho de informação e é escolhido como atributo teste.

Exercício

- Construa a árvore de decisão para esse conjunto de exemplos usando o algoritmo ID3. Indique os seus cálculos.

Pessoa	Cabelo (em cm)	Peso (em kg)	Idade	Classe
P1	0	125	36	masculino
P2	25	75	34	feminino
P3	5	45	10	masculino
P4	15	39	8	feminino
P5	10	10	1	feminino
P6	2,5	85	70	masculino
P7	20	80	41	feminino
P8	25	90	38	masculino
P9	15	100	45	masculino

Leituras

- MITCHELL, T. Machine Learning. McGraw Hill, 1997
  - Capítulo 3: Decision Tree Learning.
- REZENDE, S. (Ed.) Sistemas Inteligentes - Fundamentos e Aplicações. Manole, 2003.
  - Capítulo 5: Indução de Regras e Árvores de Decisão.
- RUSSEL, S.; NORVIK, P. Artificial Intelligence: A Modern Approach. Prentice Hall, 1995.
  - Capítulo 18: Aprendizado a partir de Observações