

# Aprendizado em IA



Prof. Carlos H. C. Ribeiro

ITA - Divisão de Ciência da Computação

# Tópicos

---

Agentes baseados em aprendizado

Aprendizado indutivo

- Árvores de decisão

- Método ID3

Aprendizado em redes neurais

- Modelos de neurônios artificiais

- Arquiteturas

- Aprendizado supervisionado em RNAs

- Algoritmo backpropagation

Aprendizado por reforço

*Cerca de 50% dos artigos publicados em IA são sobre aprendizado...*

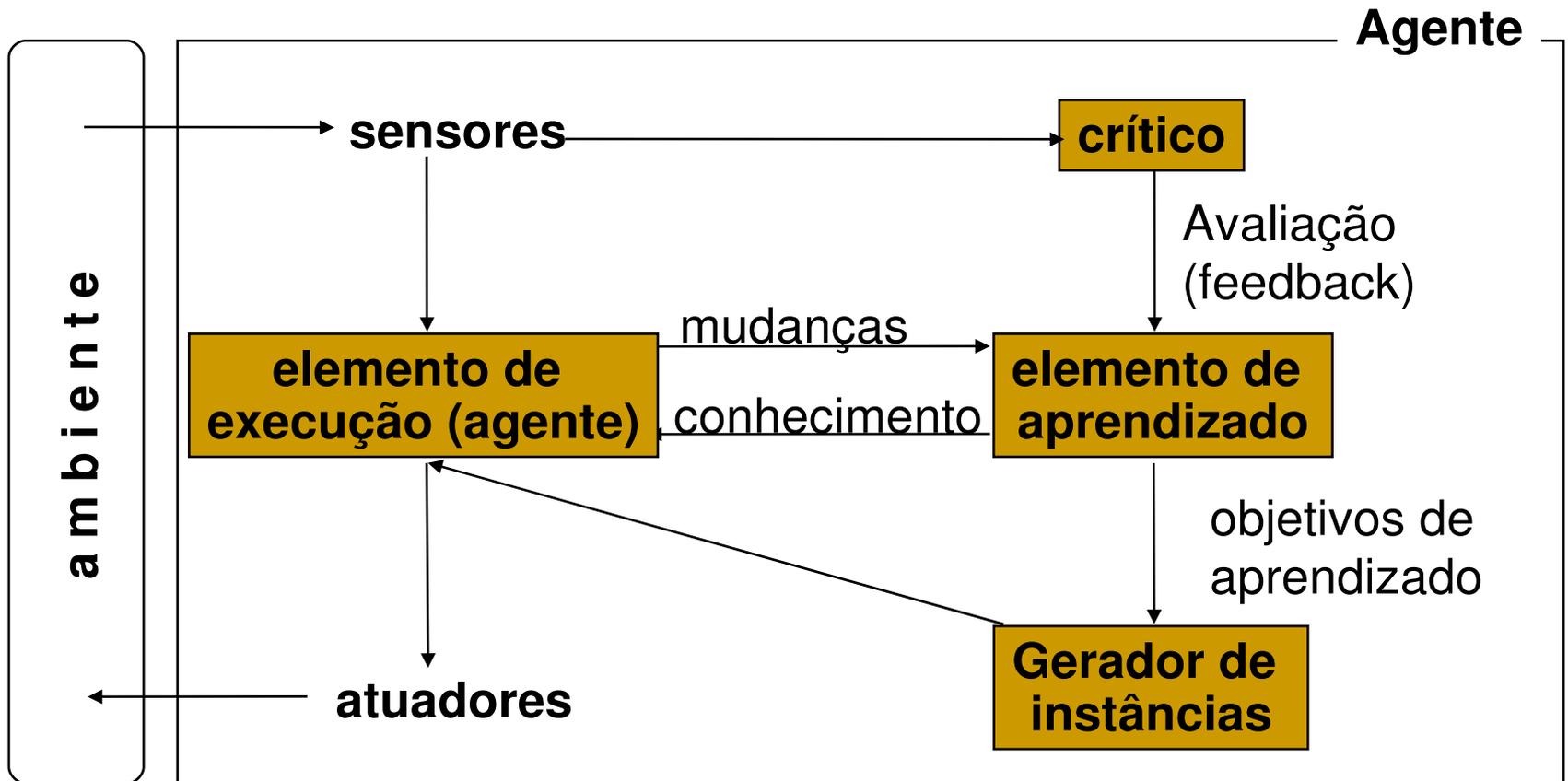
# Aprendizado – por que?

---

- ❑ Capacidade de aprender é parte fundamental do conceito de inteligência.
- ❑ Um agente aprendiz é mais flexível → aprendizado permite lidar com situações novas (mundo é dinâmico). Dá **autonomia** ao agente.
- ❑ Aprendizado facilita tarefa do projetista → programar apenas o essencial

Como construir programas (agentes) que automaticamente melhoram com a experiência?

# Agente que aprende



- **Aprendizado:** processo de modificação dos parâmetros do agente, de modo a maximizar uma medida de desempenho.

# Um Modelo Geral

---

- ❑ Ambiente / Sensores / Atuadores
- ❑ Crítico: comunica ao EA quão bem ou mal o agente está operando, de acordo com um critério fixo. Observações sensoriais nem sempre são boas indicadores...
- ❑ Elemento de Aprendizado: armazena informação sobre como a modificação dos parâmetros do EE deve ser feita (algoritmos, estruturas de dados, medida de desempenho, conhecimentos a priori, etc).
- ❑ Gerador de Instâncias: sugere ações alternativas que podem ser tomadas pelo agente, com o fim de adquirir informação adicional.
- ❑ Elemento de Execução: executa as ações de acordo com medida geral de desempenho.

# Aprendizado - paradigmas

---

## □ **Aprendizado supervisionado**

- O crítico comunica a EA o erro relativo entre a ação que deve ser tomada idealmente pelo EE e a ação efetivamente escolhida pelo agente. Pares (corretos) de entrada/saída podem ser observados (ou demonstrados por um supervisor).

## □ **Aprendizado não-supervisionado**

- O crítico não envia nenhum tipo de informação ao EA, não há “pistas” sobre as saídas corretas (geralmente utiliza-se regularidades, propriedades estatísticas dos dados sensoriais).

## □ **Aprendizado por reforço**

- O crítico comunica apenas uma indicação de desempenho (geralmente, indicação de quão bom ou ruim é o estado resultante), normalmente de modo intermitente e apenas quando situações dramáticas são atingidas (*feedback* indireto, com retardo).

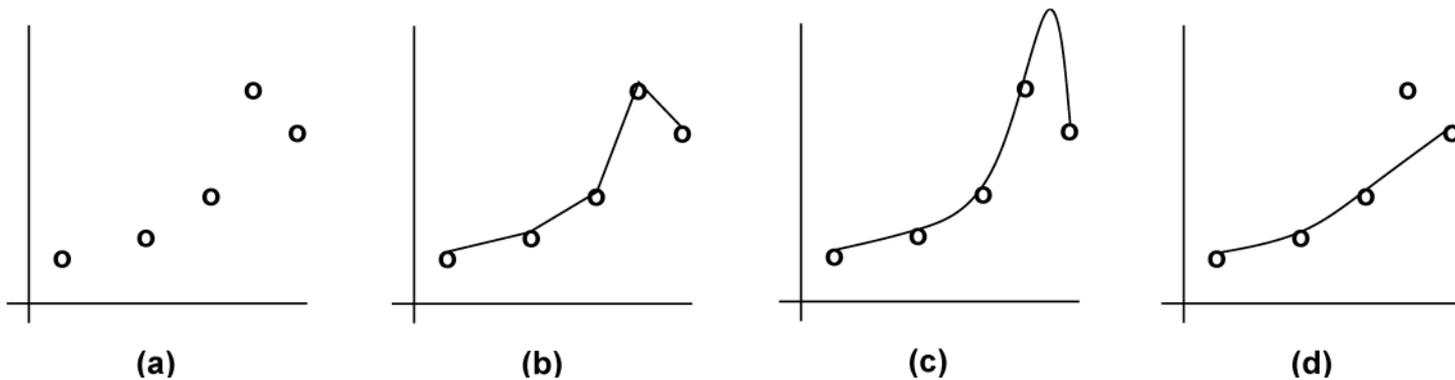
# Aprendizado Indutivo

---

- Paradigma: aprendizado supervisionado.
- Inferência de uma regra geral (hipótese) a partir de exemplos particulares (generalização).
- Precisão diretamente proporcional à quantidade de exemplos. Abordagens:
  - **não-incremental**: gera hipótese a partir de todo o conjunto de exemplos
    - eficiente, conceitualmente simples... Porém não é aplicável a muitos problemas práticos!
  - **incremental**: atualiza hipótese a cada novo exemplo
    - mais flexível, algoritmos *anytime*... Porém a ordem de apresentação é importante!

# Aprendizado Indutivo

- $x$ : entrada;  $f(x)$ : saída desejada
- Exemplo (par de treinamento) =  $(x, f(x))$
- Objetivo: aprender uma função  $h$  (hipótese) que aproxime  $f$ .



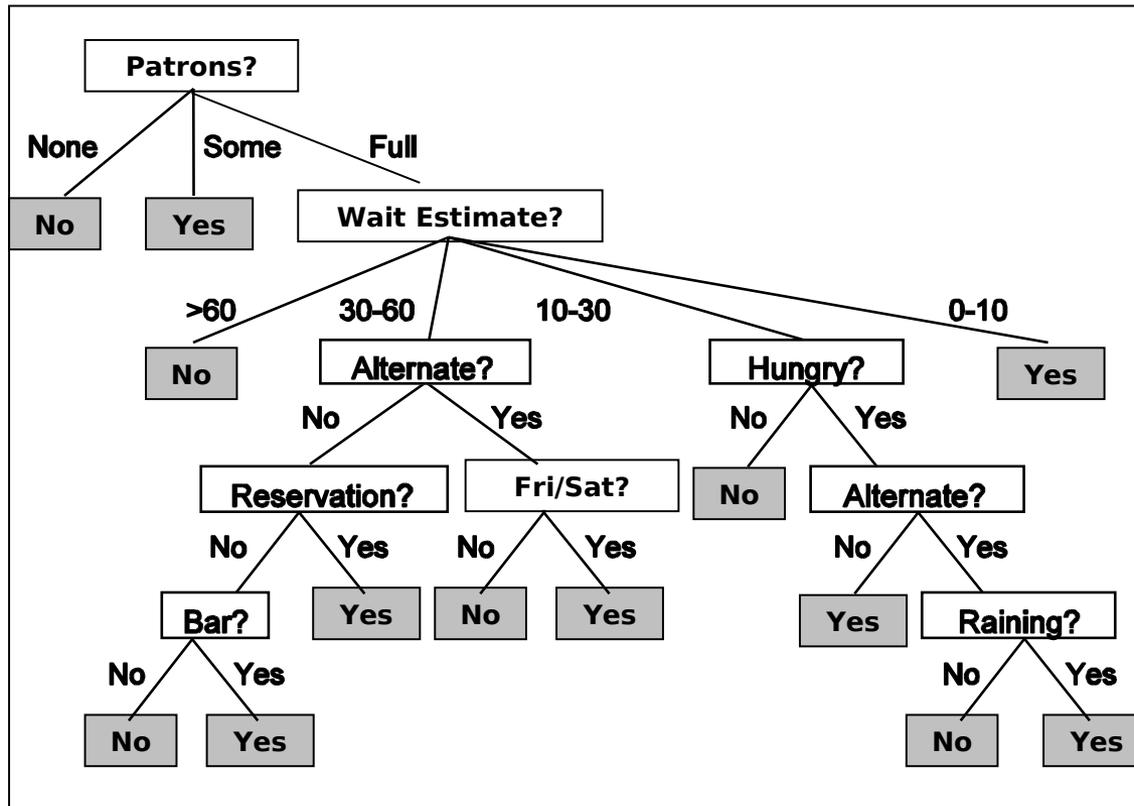
# Árvores de Decisão

---

- Entrada: situação descrita por um conjunto de propriedades;
- Saída: “decisão” SIM / NÃO (existem variantes)
- Função aprendida: representada por uma árvore de decisão (ou conjunto de regras IF-THEN)
- Árvore de decisão representa funções booleanas
  - Nós: testes para propriedades.
  - Ramos: valores dos testes.

# Árvore de Decisão: Exemplo

- Problema: esperar por uma mesa vazia?



# Aprendizado de Árvores de Decisão

Example	Attributes										Goal
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
<i>X<sub>1</sub></i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>0±10</i>	<i>Yes</i>
<i>X<sub>2</sub></i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>30±60</i>	<i>No</i>
<i>X<sub>3</sub></i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Some</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>0±10</i>	<i>Yes</i>
<i>X<sub>4</sub></i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>10±30</i>	<i>Yes</i>
<i>X<sub>5</sub></i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>&gt;60</i>	<i>No</i>
<i>X<sub>6</sub></i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Italian</i>	<i>0±10</i>	<i>Yes</i>
<i>X<sub>7</sub></i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>0±10</i>	<i>No</i>
<i>X<sub>8</sub></i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Thai</i>	<i>0±10</i>	<i>Yes</i>
<i>X<sub>9</sub></i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>&gt;60</i>	<i>No</i>
<i>X<sub>10</sub></i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>Italian</i>	<i>10±30</i>	<i>No</i>
<i>X<sub>11</sub></i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>0±10</i>	<i>No</i>
<i>X<sub>12</sub></i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>30±60</i>	<i>Yes</i>

- Um exemplo = valores de atributos, objetivo.
- Objetivo positivo/negativo: exemplo positivo/negativo
- Idéia “ingênuas”: construir árvore com um caminho para cada exemplo. Mas e se aparecer um outro exemplo?

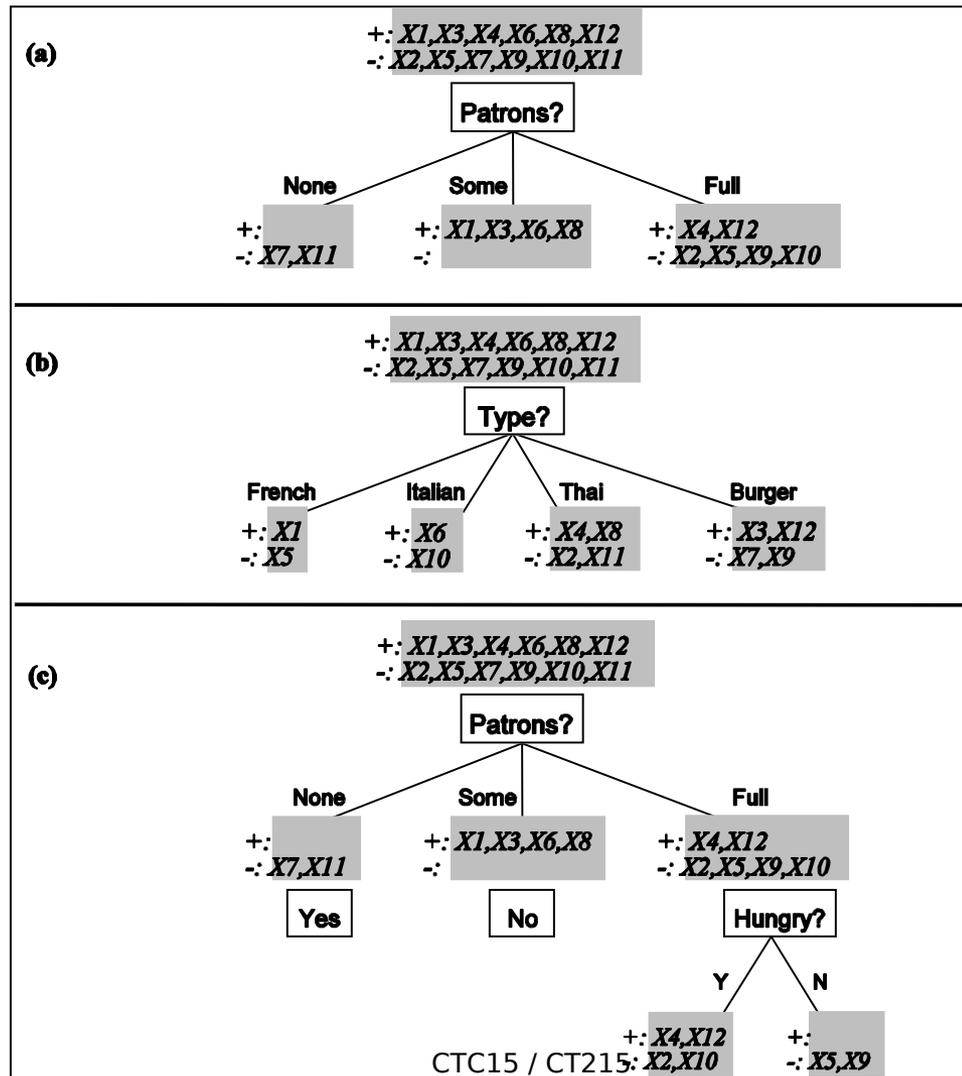
**Memorização de observações → generalização inexistente!**

# Princípio da Faca de Occam

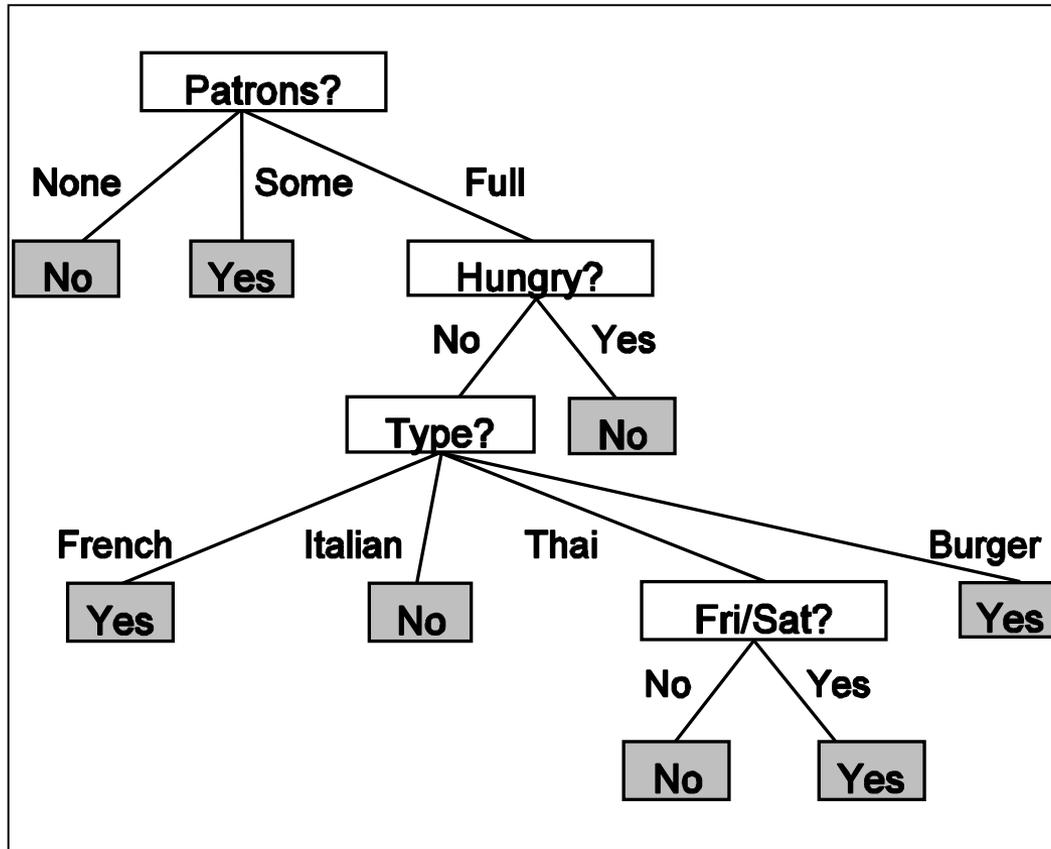
---

- A melhor hipótese é a mais simples entre aquelas consistentes com os pares de treinamento.
- Implementação recursiva:
  - Testo primeiro atributo que permita decisão imediata. Caso este não exista, testo aquele com maior poder de categorização, ou de acordo com algum outro critério, e divido a coleção de exemplos entre positivos e negativos para o atributo.
  - Exemplos restantes todos positivos (ou negativos): fim.
  - Nenhum exemplo restante: uso classificação da maioria dos exemplos do nó pai.
  - Exemplos restantes, e nenhum atributo restante: exemplos com a mesma descrição e com classificações diferentes. Dados imprecisos (ruído, atributos ocultos).

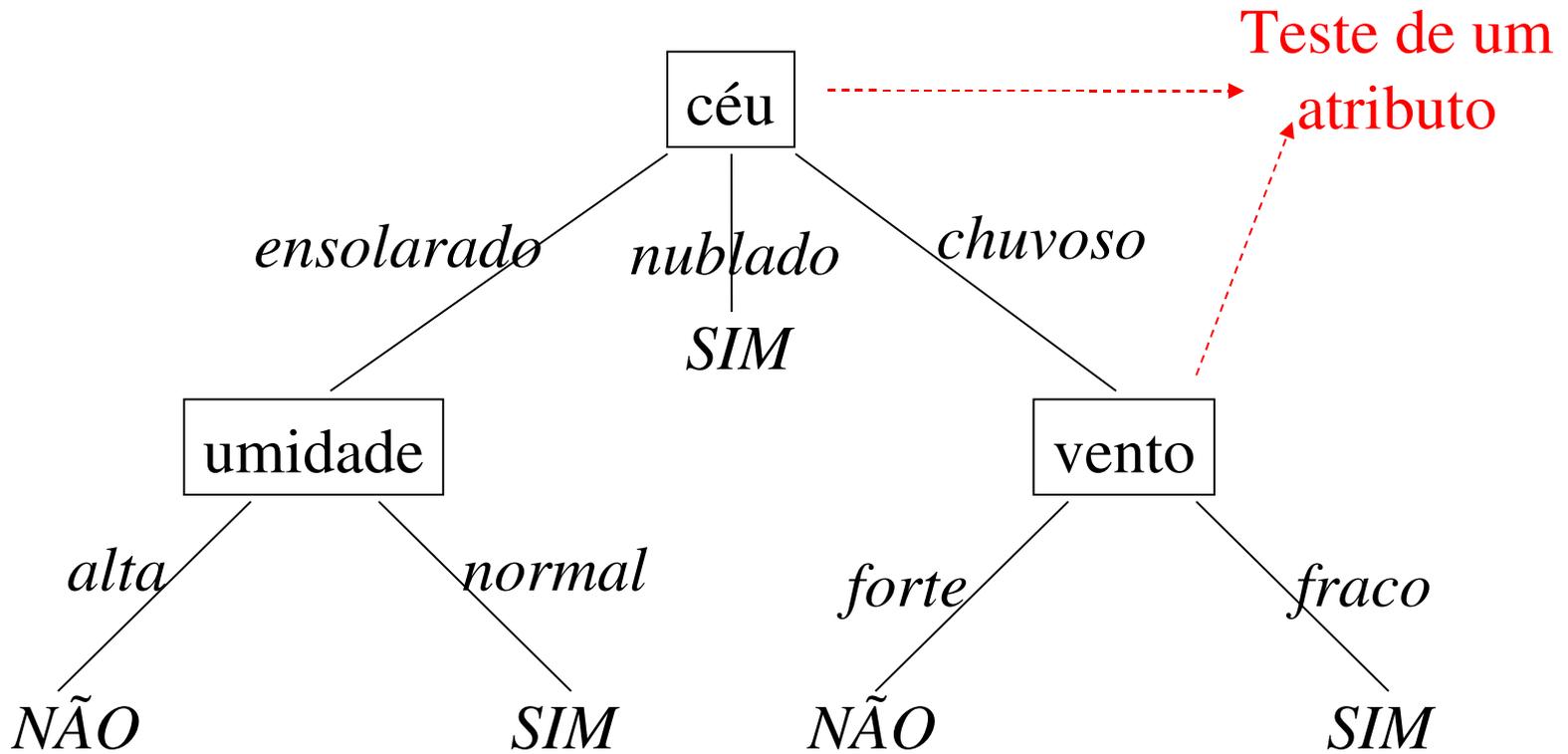
# Dividindo coleção de eventos: exemplos



# Árvore de decisão induzida



# Conceito a aprender: Devo jogar tênis?



# Árvore: disjunção de conjunções

---

□ Devo jogar tênis quando:

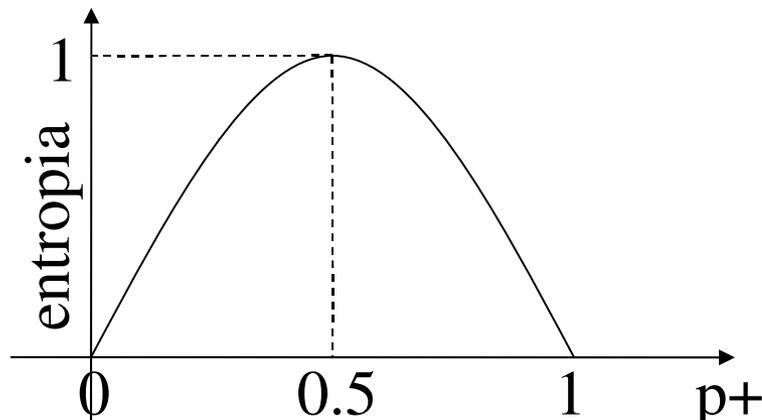
(céu = ensolarado  $\wedge$  umidade = normal)

∨ (céu = nublado)

∨ (céu = chuvoso  $\wedge$  vento = fraco)

# Qual atributo é o melhor classificador?

- Medida baseada em **ganho de informação**, calculado pela **entropia**
  - **Entropia**: medida de “impureza” numa coleção de exemplos de treinamento  $S$ 
    - Entropia = 0: todos membros da mesma classe
    - Entropia = 1: coleção com mesmo número de + e -
- $$\text{Entropia}(S) = - (p+) \log_2 (p+) - (p-) \log_2 (p-)$$



$p+$ : proporção de exemplos positivos em  $S$

# Entropia

---

$$\text{Entropia}(S) = - (p+) \log_2 (p+) - (p-) \log_2 (p-)$$

- Ex: se S tem 14 exemplos, sendo 9 positivos e 5 negativos, vem:

$$\begin{aligned} \text{Entropia}(S) = & - (9/14) \log_2 (9/14) - \\ & - (5/14) \log_2 (5/14) = 0.940 \end{aligned}$$

- Se atributo pode ter c-valores, então:

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

*OBS:  $0 \log_2 0 = 0$*

# Ganho de Informação

---

- Mede a redução esperada na entropia, causada pela partição nos exemplos segundo um atributo.
- $\text{Ganho}(S,A)$ : ganho de informação de um atributo  $A$ , relativo à coleção de exemplos  $S$ .

$$\text{Ganho}(S,A) = \text{entropia}(S) - \sum_{v \in \text{valores}(A)} (|S_v| / |S|) \text{entropia}(S_v)$$

$S_v$ : subconjunto de  $S$  no qual  $A$  tem valor  $v$

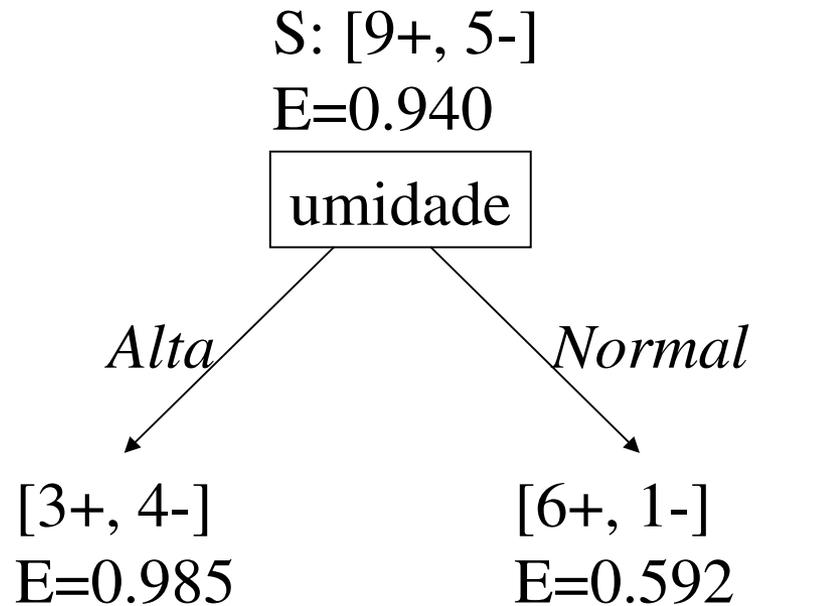
$\text{valores}(A)$ : todos possíveis valores do atributo  $A$

$$S_v = \{s \in S \mid A(s) = v\}$$

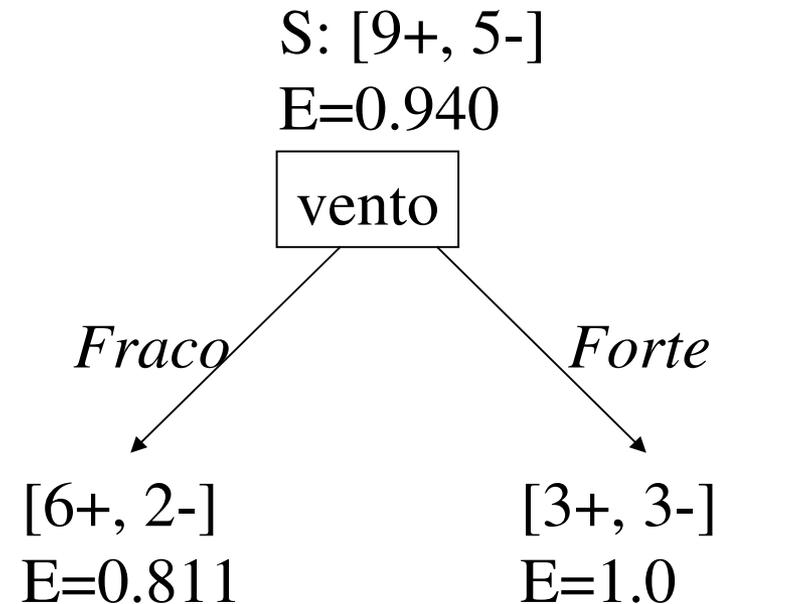
# Exemplos de treinamento para o atributo-alvo *JogarTênis*

<b>Ex</b>	<b>Céu</b>	<b>Temperatura</b>	<b>Umidade</b>	<b>Vento</b>	<b>JogarTênis</b>
X1	Ensolarado	Quente	Alta	Fraco	NÃO
X2	Ensolarado	Quente	Alta	Forte	NÃO
X3	Nublado	Quente	Alta	Fraco	SIM
X4	Chuvoso	Boa	Alta	Fraco	SIM
X5	Chuvoso	Fria	Normal	Fraco	SIM
X6	Chuvoso	Fria	Normal	Forte	NÃO
X7	Nublado	Fria	Normal	Forte	SIM
X8	Ensolarado	Boa	Alta	Fraco	NÃO
X9	Ensolarado	Fria	Normal	Fraco	SIM
X10	Chuvoso	Boa	Normal	Fraco	SIM
X11	Ensolarado	Boa	Normal	Forte	SIM
X12	Nublado	Boa	Alta	Forte	SIM
X13	Nublado	Quente	Normal	Fraco	SIM
X14	Chuvoso	Boa	Alta	Forte	NÃO

# Exemplo: qual é o melhor atributo classificador?



$$\begin{aligned} \text{Ganho}(S, \text{umidade}) &= \\ &= .940 - (7/14) \cdot .985 - (7/14) \cdot .592 \\ &= 0.151 \end{aligned}$$



$$\begin{aligned} \text{Ganho}(S, \text{vento}) &= \\ &= .940 - (8/14) \cdot .811 - (6/14) \cdot 1.0 \\ &= 0.048 \end{aligned}$$

ID3: não-incremental - analisa todos os exemplos para decidir atributo classificador

# Construção da árvore de decisão com ID3

Ganho(S,céu) = 0.246; Ganho(S,umidade) = 0.151

Ganho(S,vento) = 0.048; Ganho(S,temperatura) = 0.029

→ Para S, céu é melhor!

[X1, X2, ..., X14]

[9+, 5-]

céu

*ensolarado*

*nublado*

*chuvoso*

[X1,X2,X8,X9,X11]

[X3,X7,X12,X13]

[X4,X5,X6,X10,X14]

[2+, 3-]

[4+, 0-]

[3+, 2-]

?

**SIM**

?

# Características do ID3

---

- Preferência por árvores pequenas:
  - sua busca no espaço de hipóteses aumenta a árvore somente até o tamanho necessário para classificar o conjunto de exemplos de treinamento disponível.
- Coloca mais perto da raiz aqueles atributos que oferecem o maior ganho de informação.

# Problemas gerais

---

- Estratégia de aumentar a árvore o mínimo necessário pode trazer problemas quando:
  - Há ruído nos dados;
  - Número de exemplos de treinamento é pequeno (não representativo da função buscada)
- Problema: ruído nos dados
  - Ex: dois ou mais exemplos com mesma descrição (em termos dos atributos), mas classificação diferente.
  - Soluções possíveis: (i) cada folha é rotulada com a classificação majoritária, (ii) folhas indicam probabilidade de ocorrência de cada classificação (relativo à frequência da classificação).

# Overfitting (super-especialização)

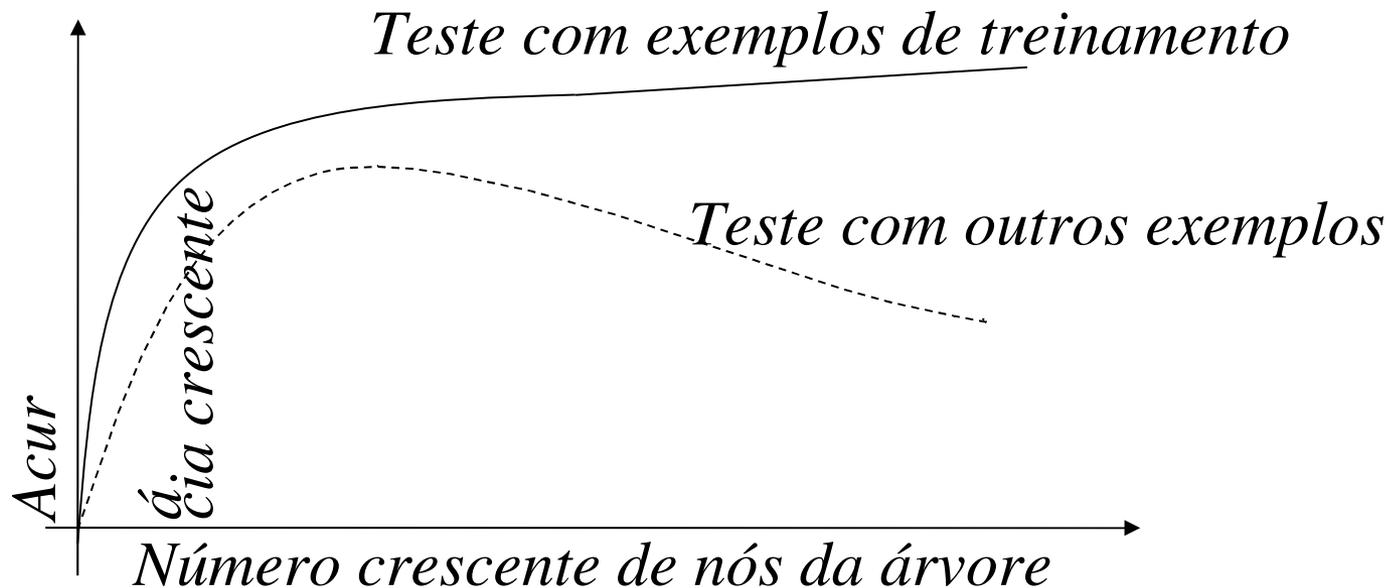
---

→ problema em todos algoritmos de aprendizado!!

**Definição:** dado um espaço de hipóteses  $H$ , uma hipótese  $h \in H$  super-especializa (**overfits**) os dados de treinamento se existir uma outra hipótese  $h' \in H$ , tal que  $h$  tem menor erro que  $h'$  no conjunto de treinamento, mas  $h'$  tem um menor erro que  $h$  sobre a distribuição total de instâncias (incluindo instâncias fora do conjunto de treinamento).

- *Como detectar atributos irrelevantes?*
- *Quão grande deve ser o ganho de informação para que o correspondente atributo seja um nó na árvore?*

# Impacto do *overfitting* num aprendizado por árvore de decisão:



Conforme ID3 adiciona mais nós para crescer a árvore de decisão, a acurácia da árvore, medida sobre os exemplos de treinamento, cresce monotonicamente. Entretanto, quando medida sobre um conjunto de dados independente do conjunto de treinamento, a acurácia primeiro cresce e, depois, decresce.

# Overfitting – Solução 1

---

- Parar de crescer a árvore antes de alcançar o ponto de classificação perfeita dos exemplos de treinamento.
  - OK, mas: quando parar?

**Validação cruzada:** tenta estimar quão bem a hipótese corrente irá predizer dados ainda não recebidos (“vistos”).

# Validação Cruzada

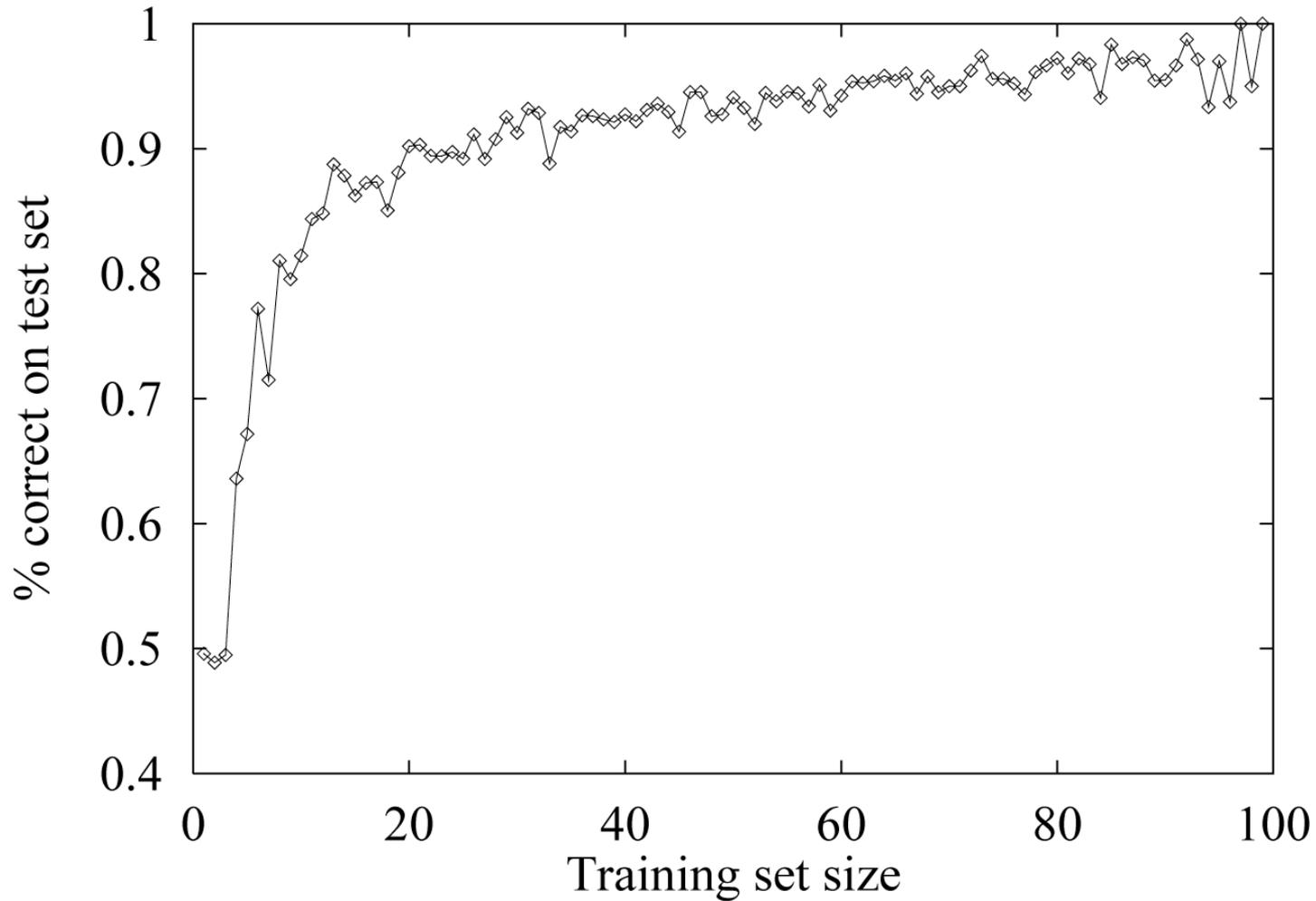
---

## □ Algoritmo



- 1) Divide o conjunto de exemplos em dois sub-conjuntos: conjuntos de treinamento (CT) e de validação (CV)
  - 2) Usa indução para gerar hipótese  $H$  sobre CT
  - 3) Mede percentagem de erro de  $H$  aplicada à CV
  - 4) Repete passos 1-3 com diferentes tamanhos de CV e CT, e tendo elementos escolhidos aleatoriamente
- Pode-se calcular a média com os dados resultantes e determinar a curva de aprendizado para o domínio em questão. Deseja-se que a qualidade da predição cresça com o tamanho do conjunto de treinamento.

# Curva de Aprendizado



# Overfitting – Solução 2

---

- Abordagens que provoquem o *overfitting* e depois podam a árvore (*pruning*)
  - Método do Erro Reduzido:
    - considera cada nó como candidato a folha (elimina sub-árvore abaixo dele), com classificação a ele associada como a mais comum; o nó se torna folha (nova árvore) sempre que a acurácia da classificação não diminuir em relação à árvore original, usando o conjunto de validação.

# Aplicações

---

## □ GASOIL

- Sistema de separação de gás-óleo em plataformas de petróleo
- Construção do sistema especialista para tal projeto usaria 10 pessoas-ano (aproximadamente 2500 regras!)
- Desenvolvido em 100 pessoas-dia, usando aprendizado de árvore de decisão.

## □ Piloto automático de um Cessna

- Treinado por três pilotos
- Obteve um desempenho melhor que os três

## □ Mineração de dados

## □ Recuperação de Informação

## □ Classificação de imagens, etc.