

**PIP/CA - Programa Interdisciplinar de Pós-Graduação  
Mestrado em Computação Aplicada da UNISINOS**

**2000/2 - 3o. Trimestre - AULA 06 / FSO**

**SISTEMAS  
ADAPTATIVOS  
INTELIGENTES**

• **Professor Responsável:**

**Prof. Dr. Fernando Osório**

**E-Mail: *osorio@exatas.unisinos.br***

**Web: *http://www.inf.unisinos.br/~osorio/sistadap.html***

## **TEMAS DE ESTUDO: *Indução de Árvores de Decisão - IDT***

### ***Introdução***

- **Árvores de Decisão** - o que é uma árvore de decisão, para que serve
- **Árvores de Decisão** - construção manual
- **Árvores de Decisão** - construção de modo automático

### ***Aprendizado por Indução - Inductive Learning***

### ***Indução de Árvores de Decisão***

- **Induction of Decision Trees**
- **IDT Methods: ID3, C4.5, CART, CN2, ASSISTANT, ...**
- **Decision Tree Learning:**
  - **ID3 Algorithm**
  - **Entropia (Entropy)**
  - **Ganho de Informação (Gain) e outra medidas (Gain Ratio)**
- **Aprendizado em métodos tipo IDT**
  - **Noisy Data, Overfitting**
  - **Pruning: Data (pre-pruning), Construction, Rule/Sub-Tree (post-pruning)**
  - **Continuous attributes, Unknown attributes, Weighted attributes**
- **Knowledge Representation // Aprendizado Incremental**

### ***Discussão final***

## ÁRVORES DE DECISÃO: *Introdução*

### *Tabela de Decisão:*

Tabela  
Verdade  
(V/F)

A	B	C	Saída
V	V	V	X
V	V	F	X
V	F	V	Y
V	F	F	Y

<i>Condição</i>					<i>Ação</i>
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# ÁRVORES DE DECISÃO: Introdução

## Regras IF-THEN:

A	B	C	Saída
V	V	V	X
V	V	F	X
V	F	V	Y
V	F	F	Y

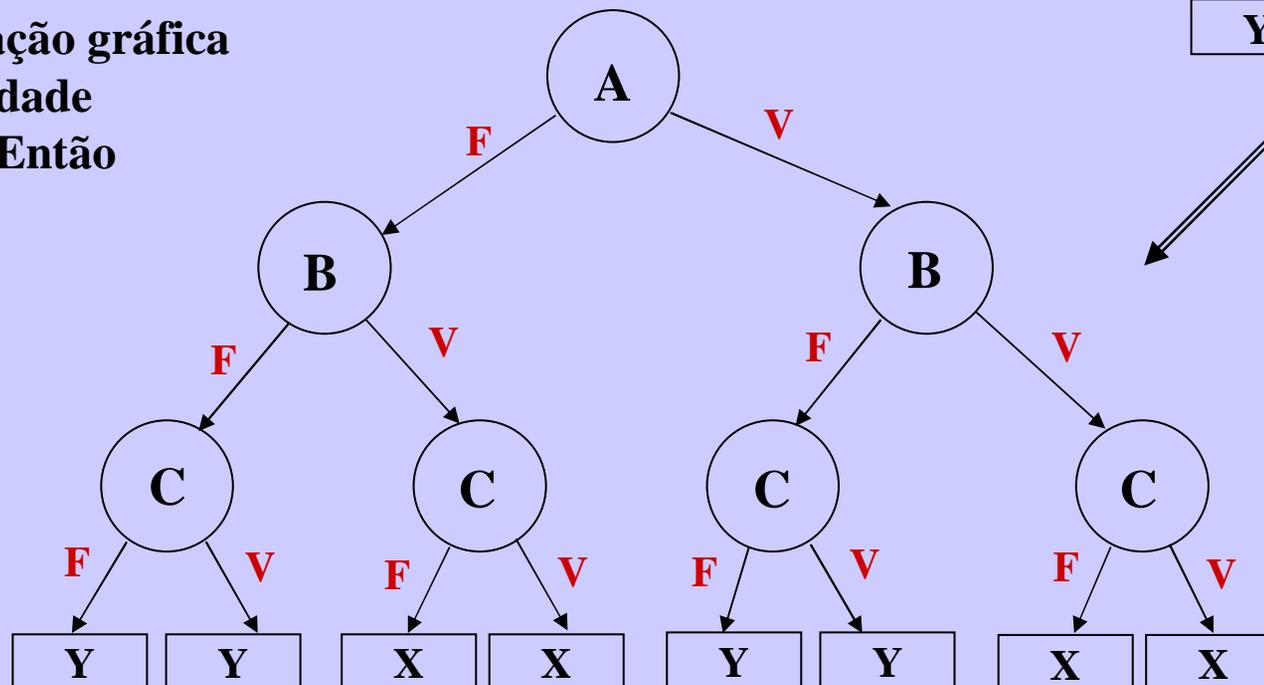
IF A and B and C THEN X  
IF A and B and not(C) THEN X  
IF A and not(b) and C THEN Y  
IF A and not(b) and not(c) THEN Y

IF B THEN X  
IF not(B) THEN Y

## Árvore de Decisão:

- Representação gráfica
- Tabela verdade
- Regras Se-Então

“construção manual”



## ÁRVORES DE DECISÃO: *Introdução*

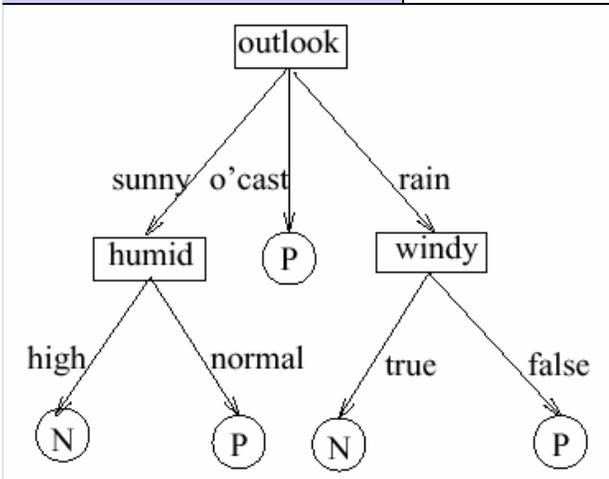
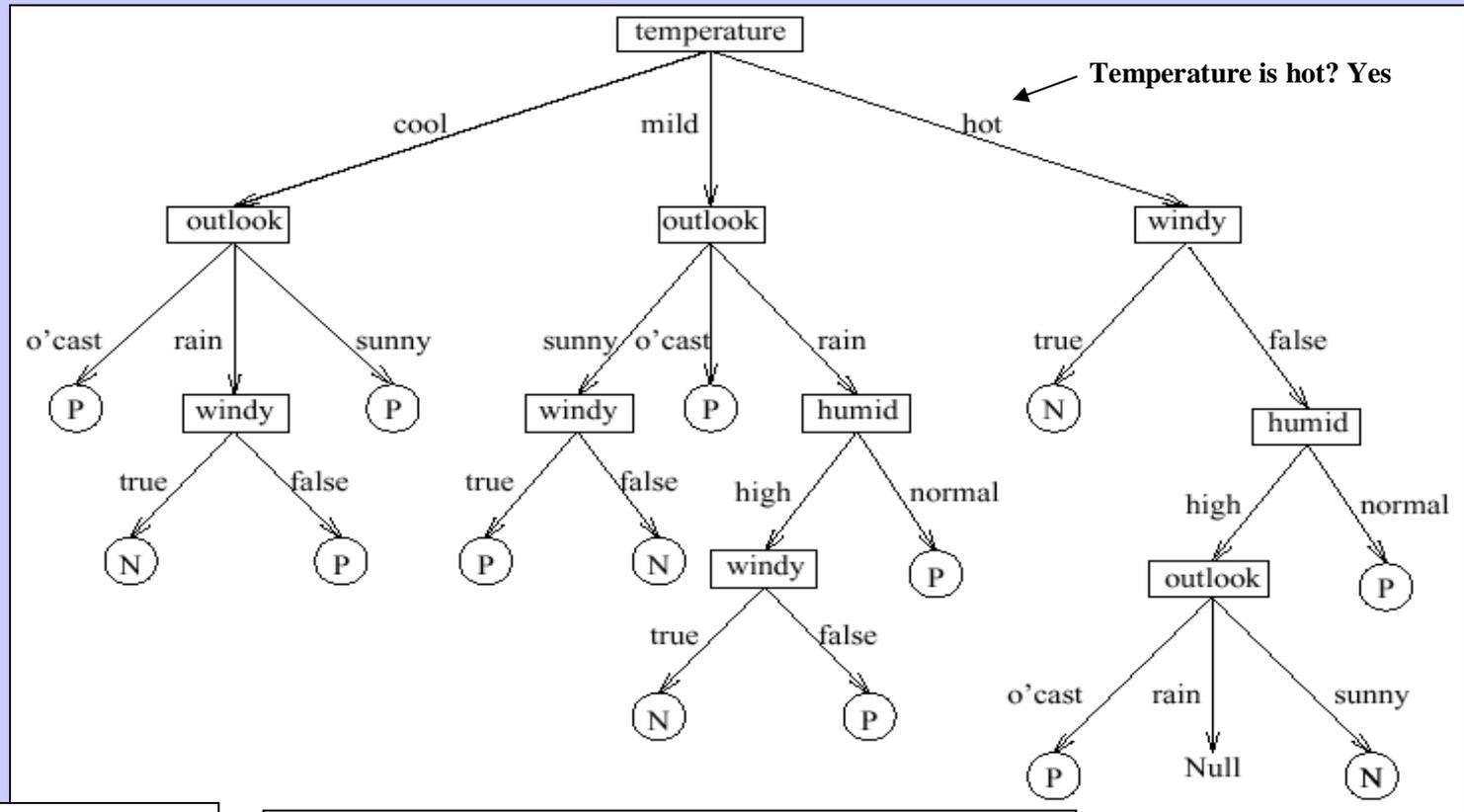
### *Aplicação das árvores de decisão:*

- Problemas onde os **conceitos** / **regras** / **ações** / **decisões** de um problema são descritos através de **exemplos** / **casos** / **instâncias**.
- As instâncias (casos) são representados por pares do tipo atributo-valor, formando uma tabela.  
Exemplo: Atributo1 = Valor1, Atributo2 = Valor2, Atributo3 = Valor3, ...
- A função objetivo tem uma saída discreta (pode ser representada por saídas binárias).  
Exemplo: Sim / Não, Classe A / Classe B / Classe C (mutuamente exclusivas)
- As árvores de decisão representam expressões disjuntivas.  
Exemplo: (A and B) ou (B and C) ou (...) ou (...) ou (...)
- Podem ser construídas de modo manual ou automático.
- Podem ter múltiplas representações (árvores diferentes) para um mesmo problema.
- Podem generalizar. Podemos ter árvores com informações redundantes ou com falta de alguma informação.
- *De um modo geral NÃO podem realizar a aproximação de funções contínuas!*

# ÁRVORES DE DECISÃO: Introdução - "existe uma árvore ideal?"

Árvores Equivalentes:

Redução do tamanho depende da seleção dos atributos (menor redundância)



Condição					Ação	
Day	Outlook	Temp.	Humidity	Wind	Play	Tennis
D1	Sunny	Hot	High	Weak	No	
D2	Sunny	Hot	High	Strong	No	
D3	Overcast	Hot	High	Weak	Yes	
D4	Rain	Mild	High	Weak	Yes	
D5	Rain	Cool	Normal	Weak	Yes	
D6	Rain	Cool	Normal	Strong	No	
D7	Overcast	Cool	Normal	Strong	Yes	
D8	Sunny	Mild	High	Weak	No	
D9	Sunny	Cool	Normal	Weak	Yes	
D10	Rain	Mild	Normal	Weak	Yes	
D11	Sunny	Mild	Normal	Strong	Yes	
D12	Overcast	Mild	High	Strong	Yes	
D13	Overcast	Hot	Normal	Weak	Yes	
D14	Rain	Mild	High	Strong	No	

## ÁRVORES DE DECISÃO: *Introdução - “generalização”*

*Sistema de auxílio ao diagnóstico de pacientes*

**Médico:** 8 perguntas sobre os sintomas do paciente (resposta: ‘S’/’N’)

**Sintomas:** 1 = Dor de Cabeça, 2 = Febre, 3 = Problemas digestivos, ...

### **Base de Conhecimentos do Médico**

1	2	3	4	5	6	7	8	Diagnóstico
S	S	N	S	N	S	S	S	Gripe
S	N	S	S	S	N	N	S	Sem problemas
S	N	S	N	S	N	S	N	Morte certa
S	N	N	S	S	N	S	N	Morte certa

**Criar uma árvore  
binária de decisão  
baseada na tabela  
de conhecimentos**

## ÁRVORES DE DECISÃO: Introdução - “generalização”

*Sistema de auxílio ao diagnóstico de pacientes*

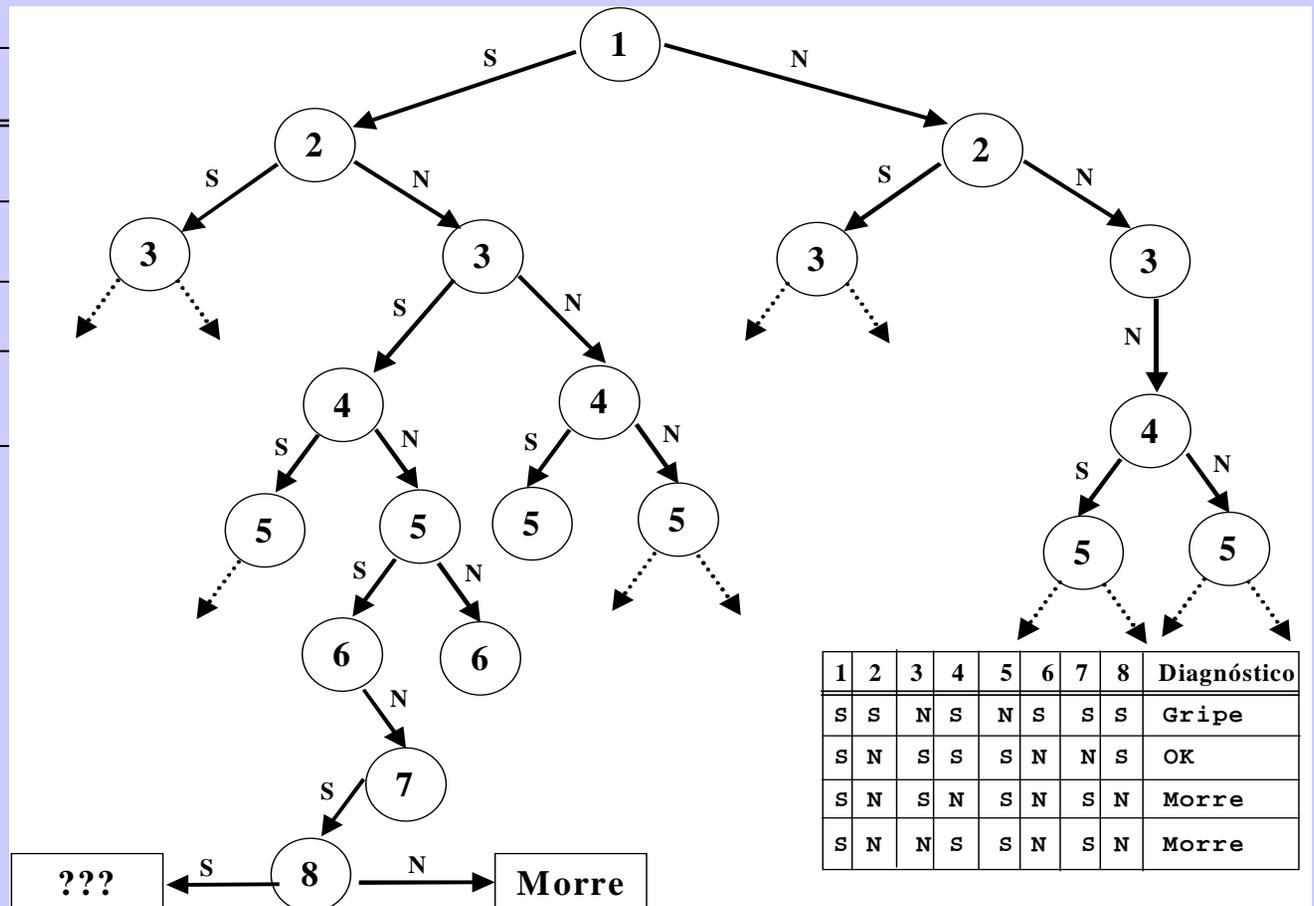
**Médico:** 8 perguntas sobre os sintomas do paciente (resposta: ‘S’/’N’)

**Sintomas:** 1 = Dor de Cabeça, 2 = Febre, 3 = Problemas digestivos, ...

**Base de Conhecimentos do Médico**

1	2	3	4
S	S	N	S
S	N	S	S
S	N	S	N
S	N	N	S

**Criar uma árvore binária de decisão baseada na tabela de conhecimentos**



## ÁRVORES DE DECISÃO: *Introdução - “generalização”*

*Sistema de auxílio ao diagnóstico de pacientes*

**Médico:** 8 perguntas sobre os sintomas do paciente (resposta: ‘S’/’N’)

**Sintomas:** 1 = Dor de Cabeça, 2 = Febre, 3 = Problemas digestivos, ...

### Base de Conhecimentos do Médico

1	2	3	4	5	6	7	8	Diagnóstico
S	S	N	S	N	S	S	S	Gripe
S	N	S	S	S	N	N	S	Sem problemas
S	N	S	N	S	N	S	N	Morte certa
S	N	N	S	S	N	S	N	Morte certa

**Criar uma árvore binária de decisão baseada na tabela**

**de conhecimentos...**

**Consultas ao sistema: (casos desconhecidos)**

S,N,S,N,S,N,S,N => Diagnóstico ?

S,N,?,?,S,N,S,N => Diagnóstico ?

## ÁRVORES DE DECISÃO: *Introdução - “árvore de decisão típica”*

- *Uso de entradas numéricas ( decisão binária usando valor numérico:  $age < 30$ ,  $age \geq 30$  )*
- *Uso de saídas discretas ( saídas binárias ou não: high, moderate, low )*

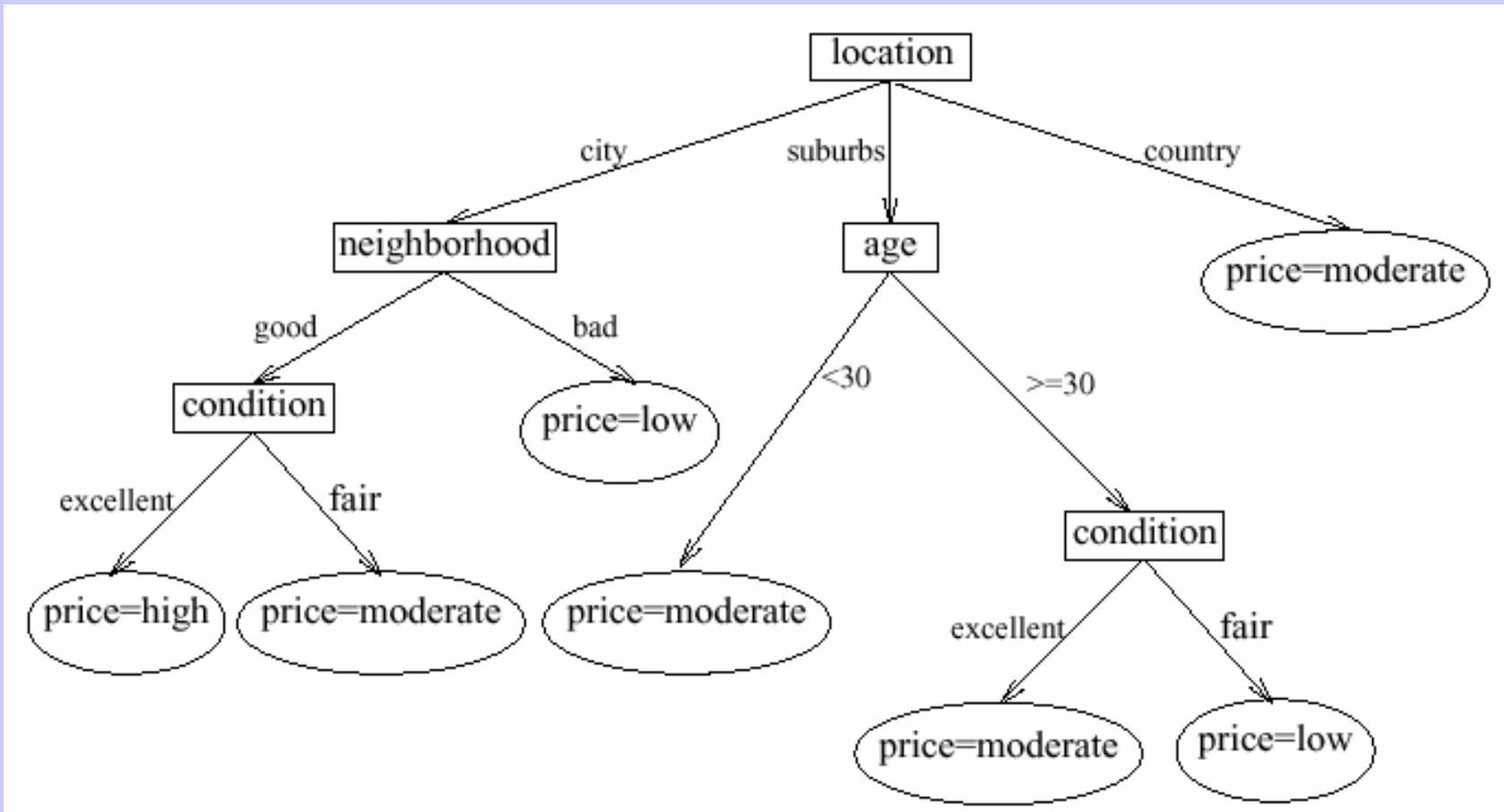


Figura - Leonard Breslow & David Aha. Simplifying Decision Trees: A Survey

## ÁRVORES DE DECISÃO: *Aprendizado por Indução - Inductive Learning*

- **Inductive Learning / Concept Learning**

- Conceitos são aprendidos a partir de um conjunto de instâncias positivas e negativas do conceito;
- O conjunto de instâncias disponíveis é dividido tipicamente em um conjunto de aprendizado e um conjunto de teste;
- Uso de algoritmos de aprendizado supervisionado: instâncias possuem classes (+/-) associada a elas;
- A saída de um algoritmo de aprendizado indutivo é usualmente dada por uma classificação ou por regras de decisão. [Nikolopoulos]

- **Concept Learning**: É o problema da busca em um espaço pré-definido de hipóteses potenciais, daquela hipótese que melhor se ajusta aos exemplos de treinamento;

- **Inductive Learning**: Toda a hipótese encontrada que se aproxime corretamente da função objetivo e que estiver baseada em um conjunto suficientemente grande de exemplos de aprendizado, também deverá aproximar corretamente esta mesma função objetivo expressa sobre um outro conjunto de exemplos não observados.

↑  
——— >> Generalização <<

*“The problem of inducing general functions from specific training examples is central to learning”* [Mitchell]

## ÁRVORES DE DECISÃO: *Indução de Árvores de Decisão*

- **Induction of Decision Trees - IDT methods**

**Conceitos:** Aprendizado de árvores de decisão é um dos métodos práticos mais usados para implementar inferência indutiva. [Mitchell]

- Este é um método para aproximação (!) de funções orientada a valores discretos, que se mostra robusto frente a dados incorretos (noisy data), sendo capaz de aprender expressões disjuntivas
- O aprendizado de árvores de decisão é um método usado para obter funções de saída discreta, no qual a função aprendida é representada por uma árvore de decisão
- Árvores de decisão também podem ser re-representadas como conjuntos de regras do tipo IF-THEN facilitando a leitura e compreensão humana.

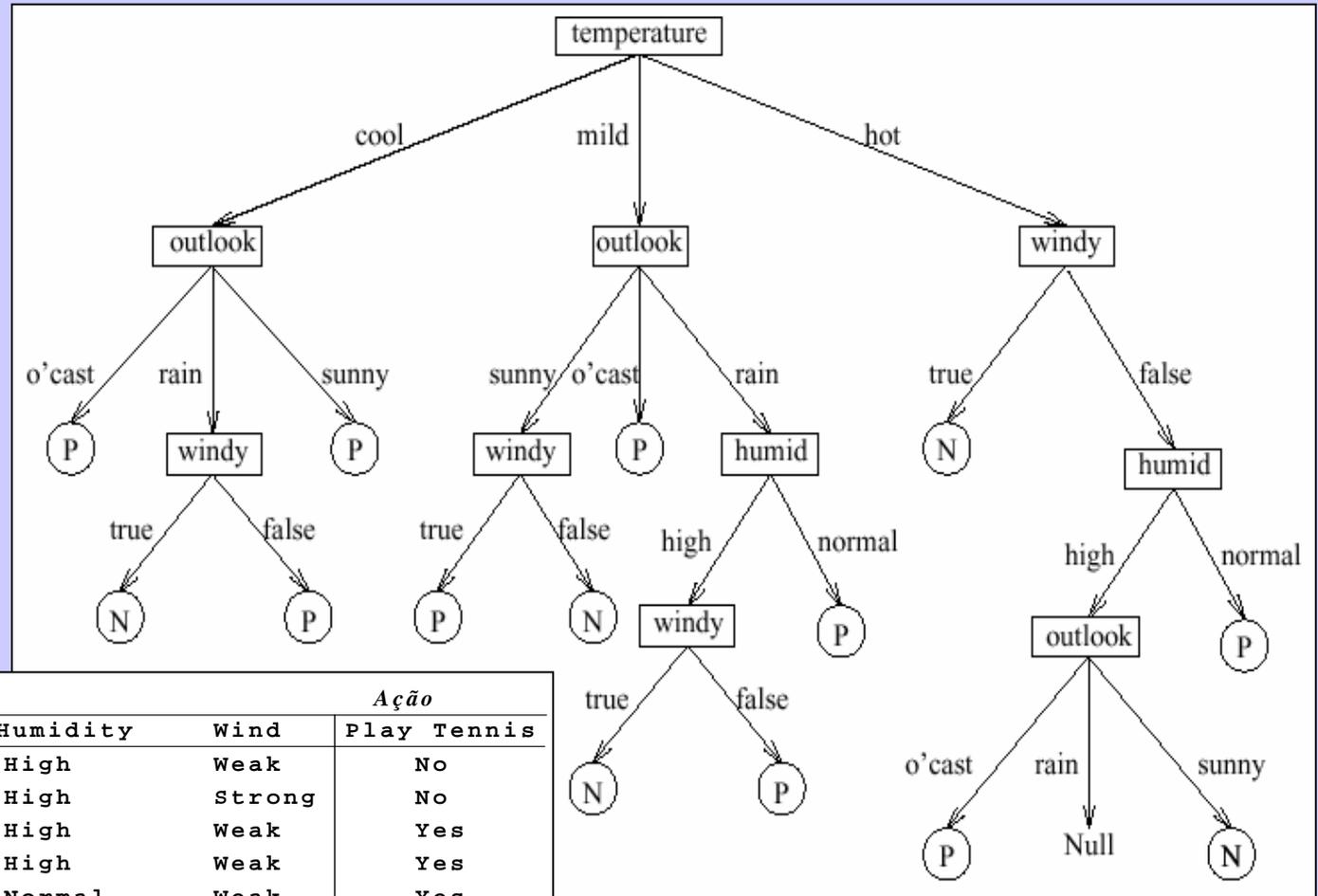
=> Questão central do problema de aprendizado de uma árvore de decisão:

*Escolher o melhor atributo para ser usado no teste de cada nodo!*

A escolha de uma **boa ordem** dos testes associados aos nodos irá gerar uma **boa árvore**: Simple, compacta e se possível com uma boa generalização [Occam's Razor].

# ÁRVORES DE DECISÃO: *Indução de Árvores de Decisão*

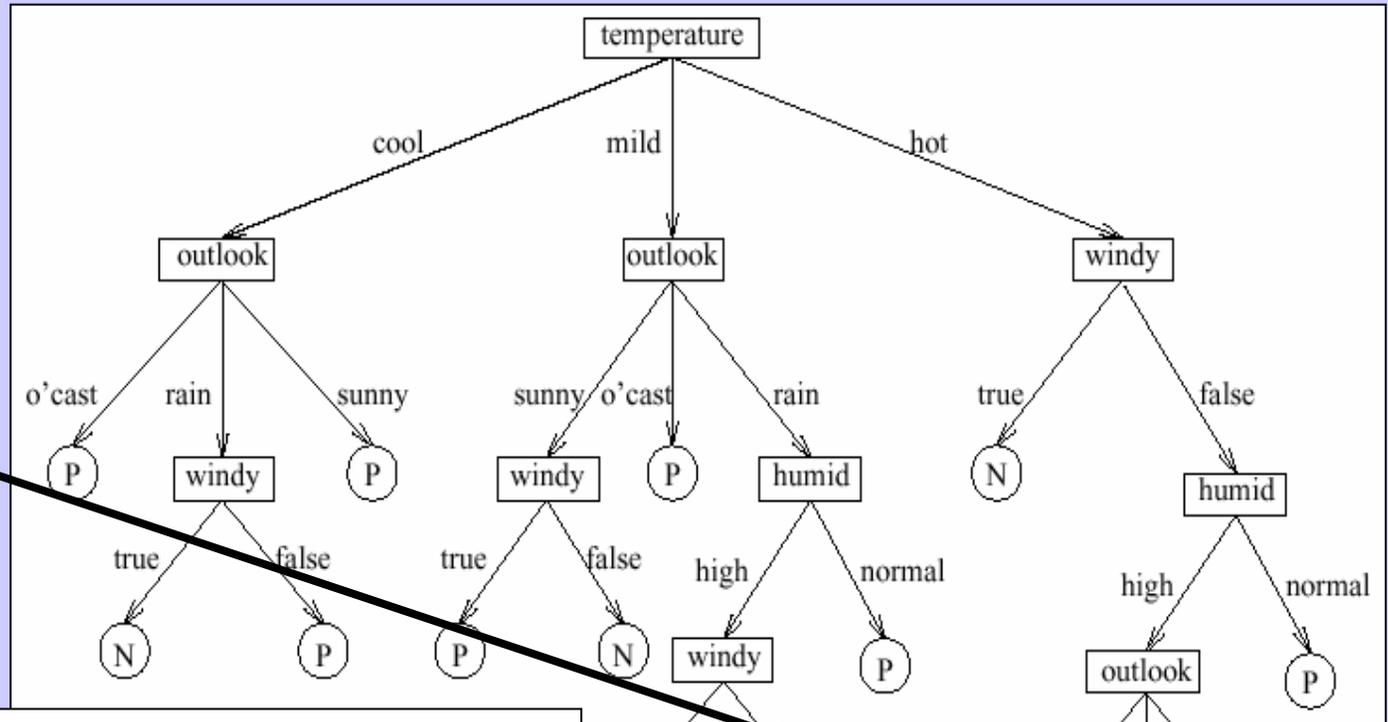
**Escolha Ruim = Árvore com muitos níveis!**



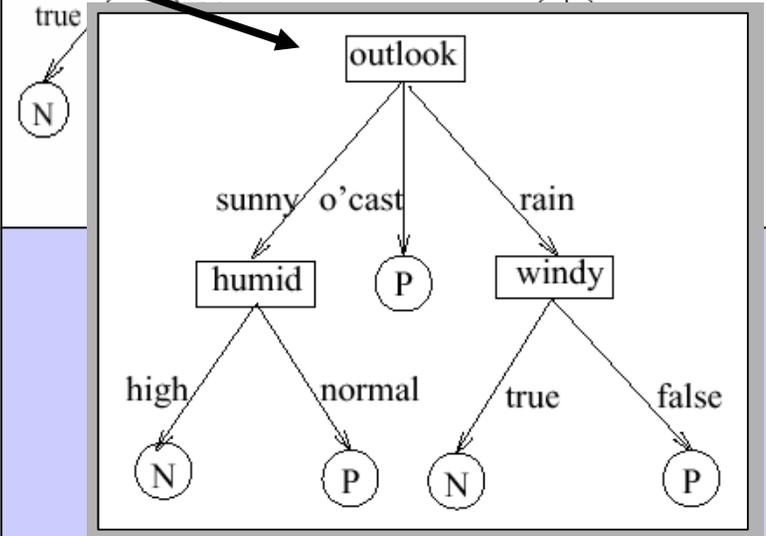
Condição					Ação
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# ÁRVORES DE DECISÃO: *Indução de Árvores de Decisão*

Escolha  
“Ótima”  
=  
Árvore mínima!



Condição					Ação
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



## ÁRVORES DE DECISÃO: *Indução de Árvores de Decisão*

### Algoritmos *IDT*:

- **ID3** - Quinlan / 1986
- **C4.5** - Quinlan / 1993 ( <http://www.cse.unsw.edu.au/~quinlan/> )
- **ASSISTANT** - Cestnik, Kononenko, Bratko / 1987
- **CART** (Classification and Regression Trees) - Breiman et al / 1984
- **CN2** - Clark, Niblett / 1987 ( <http://www.cs.utexas.edu/users/pclark/software.html> )
- **ID5R, ITI** (Incremental DT induction) - Utgoff / 1989 ( <http://www.cs.umass.edu/~lrn/iti/> )
- **Métodos similares... Família AQ** - Michalski ( <http://www.mli.gmu.edu/msoftware.html> )

### Histórico *IDT*

- Quinlan e Breiman são os personagens mais conhecidos no que diz respeito ao desenvolvimento dos algoritmos de indução de árvores de decisão;
- Os seus softwares, ID3 e CART se tornaram famosos no final da década de 80 / início 90
- O software ID3 tornou-se muito popular depois que Quinlan publicou um livro que era acompanhado deste software, na sua versão melhorada, denominado de C4.5.
- A “explosão” das aplicações de Data Mining foi fortemente influenciada pelo uso de ferramentas de Machine Learning, como o software C4.5 (e.g. Clementine)

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmo *ID3*:

- Qual é o melhor atributo? Deveria ser aquele que mais “ajuda” na classificação
- Conceito importantes: *Information Gain* e *Entropy*

#### 1. Information Gain:

Medida que indica o quanto um dado atributo irá separar os exemplos de aprendizado de acordo com a sua função objetivo (classes). Valor numérico - quantifica o ganho!  
Para determinar o ganho, precisamos calcular a “entropia” dos dados antes...

#### 2. Entropy:

Medida que indica a homogenidade dos exemplos contidos em um conjunto de dados. Permite caracterizar a “pureza” (e impureza) de uma coleção arbitrária de exemplos.

Dado o conjunto  $S$ , contendo exemplo ‘+’ e ‘-’ que definem o conceito a ser aprendido, a entropia relativa dos dados deste conjunto  $S$  é indicada por:

$$\text{ENTROPY}(S) = - P_{\oplus} \cdot \text{Log}_2 P_{\oplus} - P_{\ominus} \cdot \text{Log}_2 P_{\ominus}$$

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmo ID3: *Entropy*

$P_{\oplus}$  = Proporção entre os exemplos positivos e o total de exemplos do conjunto:  
Nro. de Casos Positivos / Nro. Total de Casos

$P_{\ominus}$  = Proporção entre os exemplos negativos e o total de exemplos do conjunto:  
Nro. de Casos Negativos / Nro. Total de Casos

Sendo que:  $0 \cdot \text{Log}_2 0 = 0$  por definição.

Exemplo: Baseado no conjunto de dados “play tennis”

$$P_{\oplus} = 9/14 \quad \text{e} \quad P_{\ominus} = 5/14$$

### 2. Entropy:

Medida que indica a homogenidade dos exemplos contidos em um conjunto de dados. Permite caracterizar a “pureza” (e impureza) de uma coleção arbitrária de exemplos.

Dado o conjunto S, contendo exemplo ‘+’ e ‘-’ que definem o conceito a ser aprendido, a entropia relativa dos dados deste conjunto S é indicada por:

$$\text{ENTROPY (S)} = - P_{\oplus} \cdot \text{Log}_2 P_{\oplus} - P_{\ominus} \cdot \text{Log}_2 P_{\ominus}$$

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmo ID3: *Entropy*

$P_{\oplus}$  = Proporção entre os exemplos positivos e o total de exemplos do conjunto:  
Nro. de Casos Positivos / Nro. Total de Casos

$P_{\ominus}$  = Proporção entre os exemplos negativos e o total de exemplos do conjunto:  
Nro. de Casos Negativos / Nro. Total de Casos

Sendo que:  $0 \cdot \text{Log}_2 0 = 0$  por definição.

Exemplo: Baseado no conjunto de dados “play tennis”

$$P_{\oplus} = 9/14 \quad \text{e} \quad P_{\ominus} = 5/14$$

$$\text{ENTROPY (S)} = - P_{\oplus} \cdot \text{Log}_2 P_{\oplus} - P_{\ominus} \cdot \text{Log}_2 P_{\ominus}$$

Logo, na base de 14 exemplos, onde 9 são positivos (Yes) e 5 são negativos (No), teremos...

$$\text{Entropy (Splay\_tennis)} = 0.94028595$$

*Relembrando...*  $\text{Log}_2 X = \text{Log } X / \text{Log } 2$

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmos *ID3*: *Entropy*

$$\text{ENTROPY (S)} = - P_{\oplus} \cdot \text{Log}_2 P_{\oplus} - P_{\ominus} \cdot \text{Log}_2 P_{\ominus}$$

Fórmula: *Valor Binário*

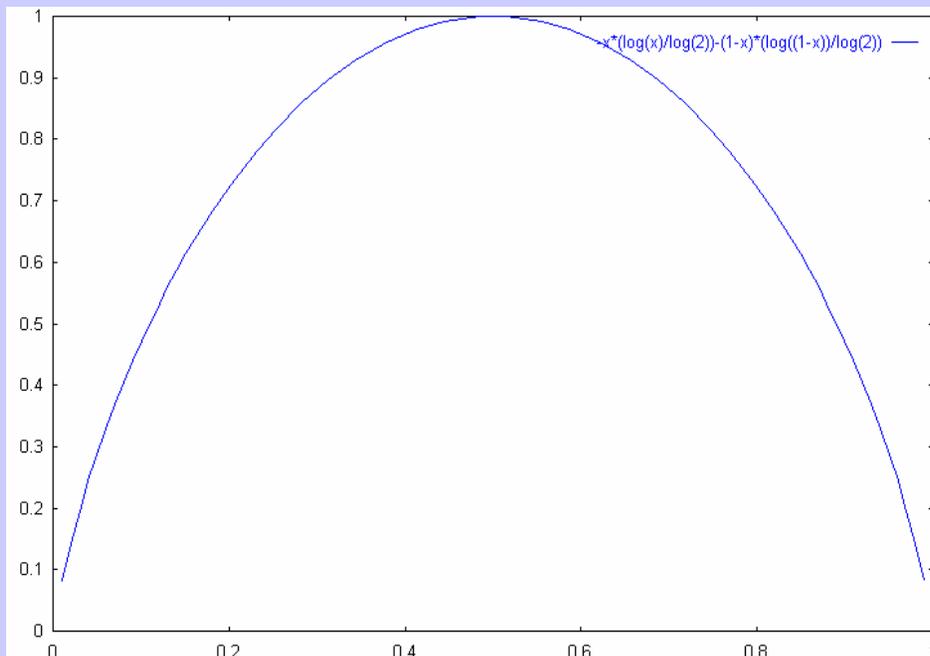
#### Play Tennis: Base de 14 exemplos

Se 9+ e 5- => Entropy (S) = - (9/14) . Log<sub>2</sub> (9/14) - (5/14) . Log<sub>2</sub> (5/14) = 0.940

Se 14+ e 0- => Entropy (S) = - (1) . Log<sub>2</sub> (1) - (0) . Log<sub>2</sub> (0) = 0

Se 0+ e 14- => Entropy (S) = - (0) . Log<sub>2</sub> (0) - (1) . Log<sub>2</sub> (1) = 0

Se 7+ e 7- => Entropy (S) = - (0.5) . Log<sub>2</sub> (0.5) - (0.5) . Log<sub>2</sub> (0.5) = 1



**Entropia** relativa a um problema de classificação binária, onde esta curva que exibimos ao lado mostra sua variação a medida que a proporção de positivos e negativos variam.

**Entropia / Shannon:** indica a qtd. de bits necessários para enviar uma mensagem dada a proporção (probabilidade) de ocorrência de um dado em relação ao outro

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmos *ID3*: *Entropy*

$$\text{ENTROPY (S)} = - P_{\oplus} \cdot \text{Log}_2 P_{\oplus} - P_{\ominus} \cdot \text{Log}_2 P_{\ominus}$$

Fórmula: *Valor Binário*

$$\text{ENTROPY (S)} = - \sum_{i=1}^N P_i \cdot \text{Log}_2 P_i$$

Fórmula: *Para "N" Classes*

De acordo com a notação do livro do Quinlan...

$$\text{INFO (S)} = - \sum_{i=1}^N \frac{\text{Freq (Ci, S)}}{|S|} \cdot \text{Log}_2 \left( \frac{\text{Freq (Ci, S)}}{|S|} \right)$$

Voltando no tempo....

A função de ganho - Information Gain - agora já pode ser calculada!

**GAIN (S, A)** = Redução esperada na entropia de S, causada pelo particionamento dos exemplos em relação a um atributo escolhido (A).

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmos *ID3: Information Gain*

**GAIN (S, A)** = Redução esperada na entropia de S, causada pelo particionamento dos exemplos em relação a um atributo escolhido (A).

$$\text{Gain (S,A)} = \text{Entropy (S)} - \sum_{v=1}^N \frac{|S_v|}{|S|} \cdot \text{Entropy (S}_v)$$

onde,

**A** = Atributo considerado

**N** = Número de valores possíveis que este atributo pode assumir

**S<sub>v</sub>** = Sub-conjunto de S onde o atributo A possui o valor V

**Exemplo:** dado o atributo Wind = { weak, strong } da tabela “play tennis”

$$\begin{aligned} \text{Gain(S, Wind)} &= \text{Entropy(S)} - ( P_{\text{Weak}} \cdot \text{Entropy}(S_{\text{Wind=Weak}}) + P_{\text{Strong}} \cdot \text{Entropy}(S_{\text{Wind=Strong}}) ) \\ &= 0.940 - ( 8/14 \cdot \text{Entropy}(S_{\text{Wind=Weak}}) + 6/14 \cdot \text{Entropy}(S_{\text{Wind=Strong}}) ) \\ &= 0.940 - ( 8/14 \cdot (-6/8 \cdot \text{Log}_2(6/8) - 2/8 \cdot \text{Log}_2(2/8)) - ( 6/14 \cdot (-3/6 \cdot \text{Log}_2(3/6) - 3/6 \cdot \text{Log}_2(3/6)) ) \\ &= 0.940 - 0.4634 - 0.4285 = \mathbf{0.048} \end{aligned}$$

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmos *ID3*: *Information Gain*

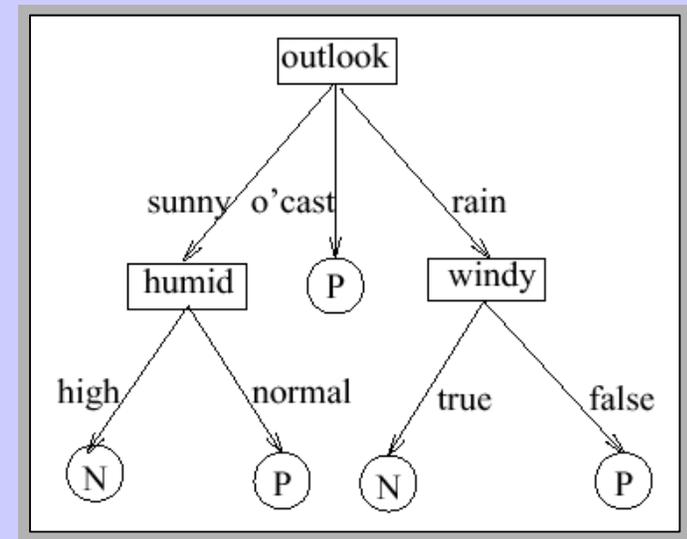
Exemplo da tabela “play tennis”

$$\text{Gain}(S, \text{Outlook}) = 0.246$$

$$\text{Gain}(S, \text{Humidity}) = 0.151$$

$$\text{Gain}(S, \text{Wind}) = 0.048$$

$$\text{Gain}(S, \text{Temperature}) = 0.029$$



Qual o melhor ganho?

Certamente a escolha do atributo “outlook” maximiza o ganho!

Exemplo “manipulado” - **Best** case (supondo uma base com 7+ e 7-)

$$\text{Gain} = 1.0 - 7/14 \cdot \text{Entropy}(\text{MagicFeature}=\text{True}) - 7/14 \cdot \text{Entropy}(\text{MagicFeature}=\text{False})$$

$$\text{Gain} = 1.0 - 7/14 \cdot -(7/7 \cdot \log_2(7/7) + 0 \cdot \log_2(0)) - 7/14 \cdot -(0 \cdot \log_2(0) + 7/7 \cdot \log_2(7/7))$$

$$\text{Gain} = 1.0 - 0.5 \cdot 0 - 0.5 \cdot 0 = 1.0 \quad (\text{Ganho MÁXIMO!})$$

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

### Algoritmos *ID3*: *Information Gain*

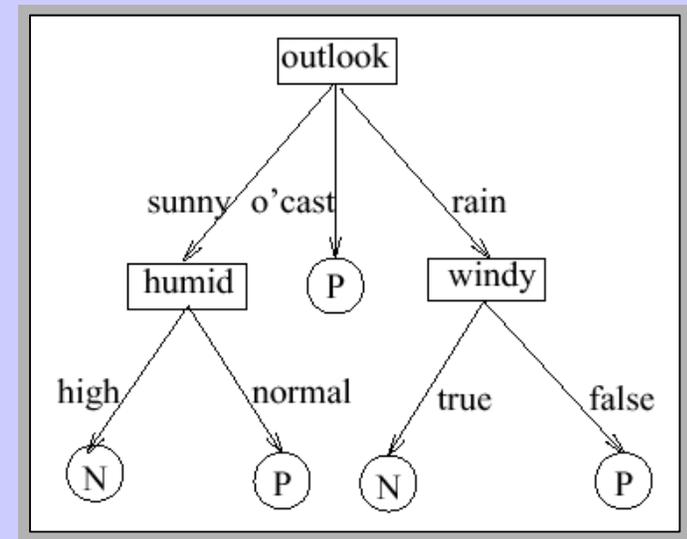
Exemplo da tabela “play tennis”

$$\text{Gain}(S, \text{Outlook}) = 0.246$$

$$\text{Gain}(S, \text{Humidity}) = 0.151$$

$$\text{Gain}(S, \text{Wind}) = 0.048$$

$$\text{Gain}(S, \text{Temperature}) = 0.029$$



Qual o melhor ganho?

Certamente a escolha do atributo “outlook” maximiza o ganho!

Exemplo “manipulado” - **Worst** case (supondo uma base com 7+ e 7-)

$$\text{Gain} = 1.0 - 6/14 \cdot \text{Entropy}(\text{NullFeature}=\text{True}) - 8/14 \cdot \text{Entropy}(\text{NullFeature}=\text{False})$$

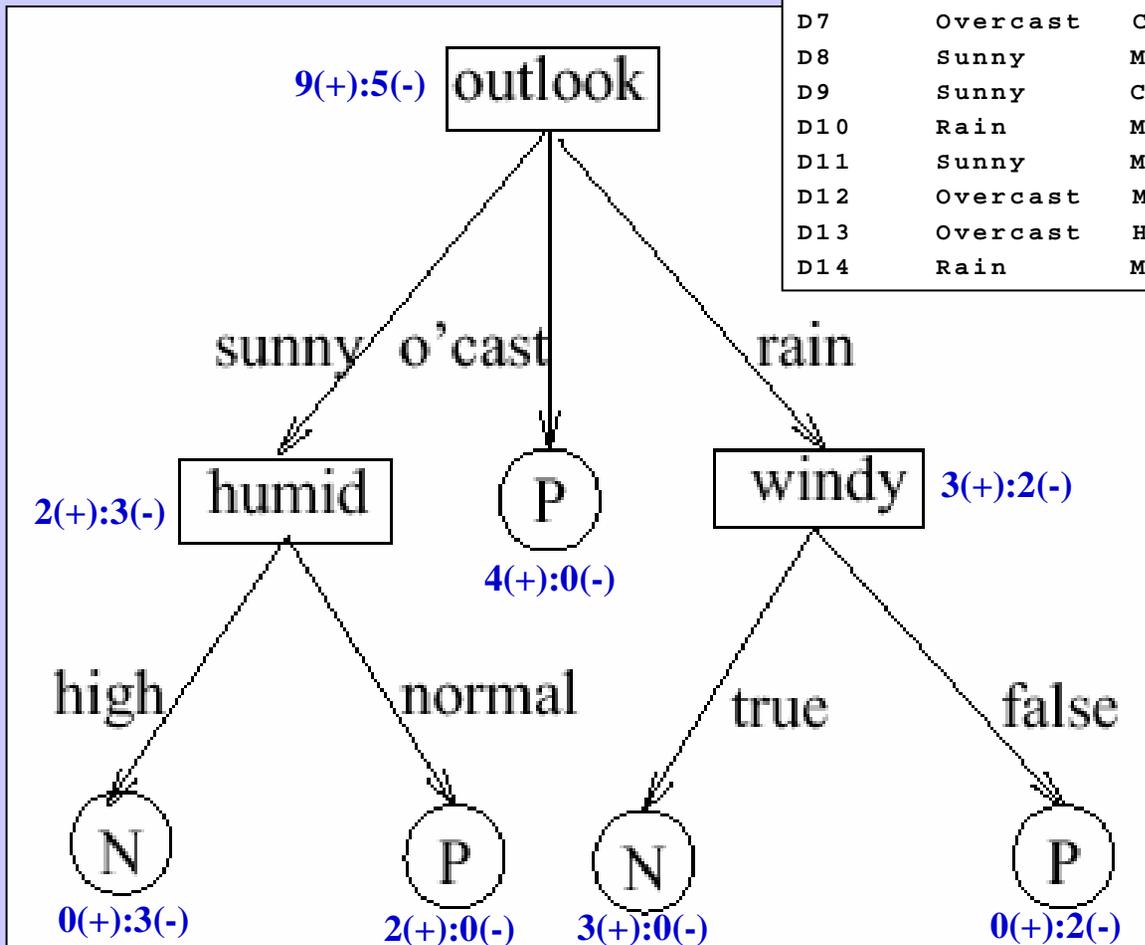
$$\text{Gain} = 1.0 - 6/14 \cdot -(3/6 \cdot \text{Log}_2(3/6) + 3/6 \cdot \text{Log}_2(3/6)) - 8/14 \cdot -(4/8 \cdot \text{Log}_2(4/8) + 4/8 \cdot \text{Log}_2(4/8))$$

$$\text{Gain} = 1.0 - 6/14 \cdot 1 - 8/14 \cdot 1 = 0.0 \quad (\text{Ganho MÍNIMO!})$$

# ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

## Algoritmos ID3: *árvore obtida*

Condição					Ação
Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão*

### Algoritmos *IDT*: Melhorias

#### - Ganho de Informação:

Quanto mais valores um atributo tiver... Maior a chance de ter um bom ganho!

Se o atributo for um identificador individual (chave única)... Ganho máximo!

Exemplo: Day - D1, D2, D3, ... Divide 1 exemplo para cada valor do atributo Day

Solução: *Gain Ratio* (C4.5) - Busca a distribuição ampla e não uniforme dos +/-.

#### - Atributos com valores numéricos:

Como dividir os intervalos usados para testar os atributos?

Exemplo: Temperatura > 38.0, Temperatura < 38.0 (usar 36, 37, 38 como limite?)

Solução: algoritmos de discretização de variáveis contínuas.

#### - Atributos com valores desconhecidos: como calcular a entropia e ganho?

#### - Atributos com importância diferente: chuva em pique-nique (só ocorre 1 vez no conj.)

#### - Generalização: se a árvore é criada até o nível das folhas (onde temos apenas os casos +, ou, apenas os casos -), então as árvores terão uma tendência a “decorar” os exemplos => *Overfitting / Noisy Data*

Solução: parar de adicionar nodos, podar a árvore, simplificar as regras.

#### - Aprendizado incremental: como aprender apenas 1 exemplo a mais? (ITI)

## ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão*

### *Algoritmos IDT: Discussão*

***Occam's Razor: Prefer the simplest hypothesis that fits the data***

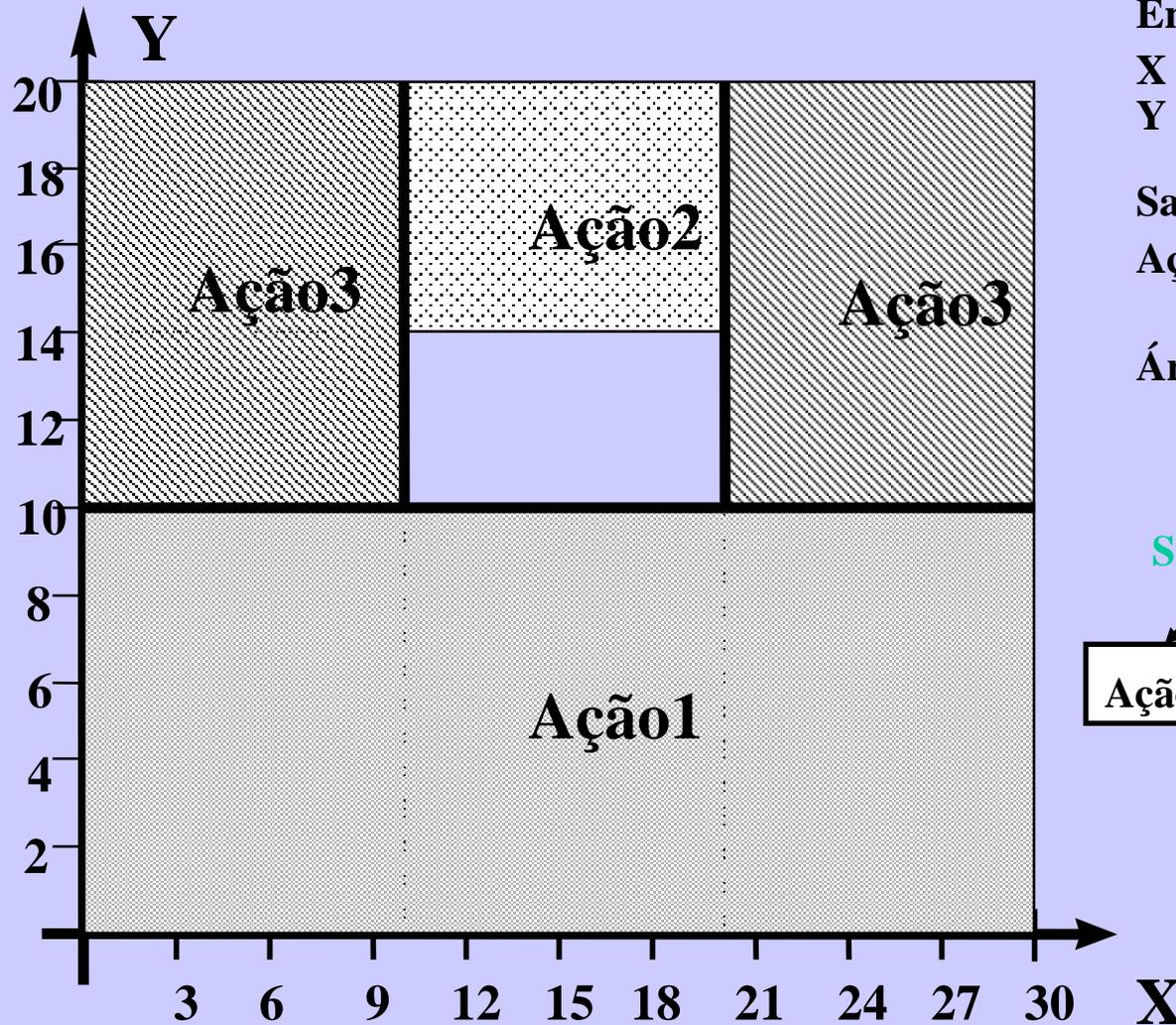
Filósofos e pensadores vem discutindo por vários séculos a questão do conhecimento e a melhor maneira de representá-lo e/ou organizá-lo. Isto nos leva ao dilema entre memorizar ou aprender (guardar a regra geral, o conceito por trás do caso particular). Willian de Occam foi um dos primeiros a discutir esta questão ao redor do ano de 1320. Desta sua discussão surgiu a máxima citada acima... OCCAM's RAZOR

Em relação as árvores de decisão (valendo também para outros métodos de aprendizado como as redes neurais!) vamos considerar o princípio de Occam como sendo nossa principal meta, e justamente a árvore mais simples deverá ser aquela que melhor deve generalizar os conceitos aprendidos.

Por isso que buscamos sempre a simplificação ao máximo das árvores, seja usando toda a teoria da informação (entropia, ganho) para dar suporte a esta busca da árvore mais simples, seja usando técnicas de poda das árvores e de regras (pre-pruning, construction pruning, post-pruning) a fim de simplificar ao máximo.

# ÁRVORES DE DECISÃO: *Aprendizado de Árvores de Decisão - Algoritmo ID3*

## Algoritmos IDT: *Representação de Conhecimento*



Entradas:

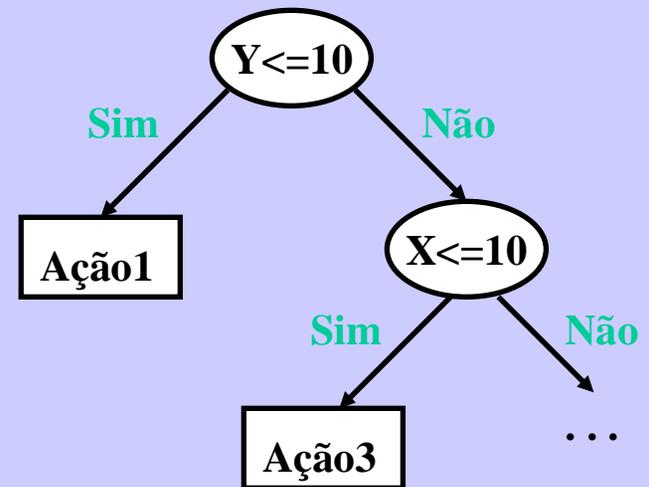
$X = [0..30]$

$Y = [0..20]$

Saídas (Ações):

Ação1  , Ação2  , Ação3 

Árvore de Decisão:





## TEMAS DE PESQUISA SOBRE REDES NEURAIS ARTIFICIAIS: MLP

### \* LIVROS / PAPERS / DOCUMENTAÇÃO SOBRE ÁRVORES DE DECISÃO:

- **Obra clássica:**

- **Quinlan, J. Ross.**

- **C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.**

- **Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.**

- **Chris Nikolopoulos. Expert Systems: Introduction to First and Second Generation and Hybrid knowledge Based Systems. Marcel Dekker Inc. 1997.**

- **Citeseer Nec: <http://www.researchindex.com/>**

### \* SOFTWARES:

- **C4.5**            <http://www.cse.unsw.edu.au/~quinlan/>

- **Dtree**            <http://fuzzy.cs.uni-magdeburg.de/~borgelt/>

- **R**                <http://www.r-project.org/>

- **CN2**            <http://www.cs.utexas.edu/users/pclark/software.html>

- **ITI**              <http://www.cs.umass.edu/~lrn/iti/>

- **AQ Family**    <http://www.mli.gmu.edu/msoftware.html>

- **CART**           <http://www.salford-systems.com/>

- **Clementine**   <http://www.isl.co.uk/clementine.html>

- <http://www.spss.com/clementine/>