

CC7711

Inteligência Artificial e Robótica

Prof. Dr. Flavio Tonidandel



The background of the slide features a complex network diagram. It consists of numerous circular nodes of varying sizes, connected by thin, light-blue lines. The nodes are distributed across the entire frame, with a higher density on the left side, creating a sense of depth and connectivity. The overall color palette is light blue and white, giving it a technical and modern appearance.

Aprendizado Indutivo

CC7711 - Inteligência Artificial e Robótica

Aprendizado

Por que aprender ?

- 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 e deixa o sistema aprender o resto sozinho
- Para permitir adaptabilidade do sistema
 - ambiente dinâmico
 - reatividade !!

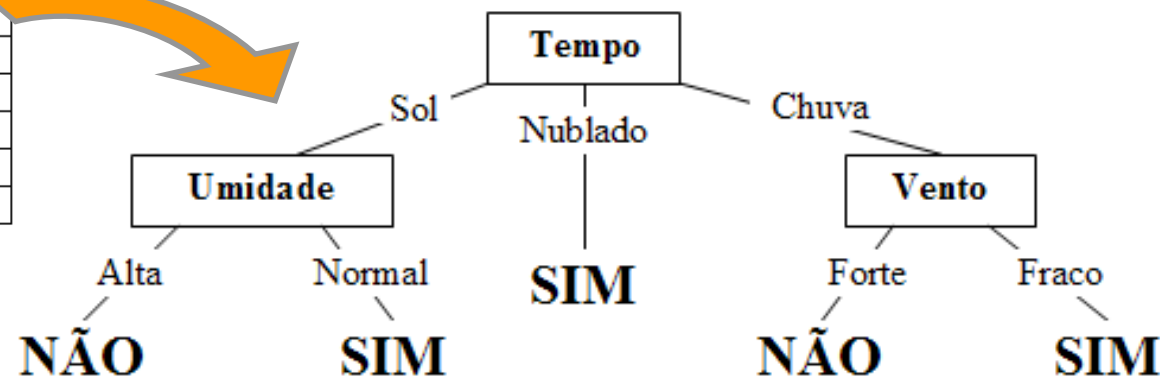
Paradigmas de Aprendizado

- Aprendizagem **supervisionada**:
 - Dado um conjunto de exemplos pré-classificados, aprender uma descrição geral que encapsula a informação contida nesses exemplos e que pode ser usada para prever casos futuros
- Aprendizagem **não-supervisionada**:
 - Dada uma coleção de dados não classificados obtidos de sensores ou por outros meios, agrupa-os por regularidades (criar clusters de informação) e deles extrai informações relevantes.
- Aprendizagem **por reforço**:
 - Trabalha por recompensa e punição

Aprendizado Indutivo

- Dado um conjunto de exemplos pré-classificados

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Momo	Alta	Forte	NÃO



... gerar uma árvore de decisão.

Aprendizado Indutivo

- Paradigma: aprendizado supervisionado.
- Características:
 - Inferência de uma regra geral (hipótese) a partir de exemplos particulares (Exemplos de Treinamento)
 - Precisão diretamente proporcional à quantidade de exemplos.
- Métodos:
 - simbólicos (ex:ID3 e C4.5);
 - não-simbólicos (ex:Redes Neurais).

Aprendizado: Sistema ID3

- O ID3 é aplicado para GENERALIZAÇÕES
- Classifica instâncias alocando-as de cima para baixo em uma árvore de decisão
- **Instância** = par *atributo-valor*.
 - Ex: Temperatura = {quente,frio,morno}
onde Temperatura é um *atributo* e {quente,frio,morno} são *valores*
- O ID3 constrói uma árvore de decisão tentando sempre responder a seguinte pergunta ?
 - Qual a melhor instância para alocar em cada nó da árvore ?

ID3 - Árvore de Decisão

- Vamos entender o ID3 por meio de um exemplo...
- Suponha que se queira decidir se haverá, dependendo das condições do dia, uma partida de tênis.
- Tem-se uma quantidade de observações dos dias que ocorreram ou não uma partida de tênis
- Queremos criar regras gerais (em forma de árvore de decisão) que permite o sistema decidir se haverá ou não uma partida de tênis
- Temos as seguintes instâncias:
 - **Temperatura** = {Quente, Frio, Morno}
 - **Umidade** = {Alta, Normal}
 - **Vento** = {Forte, Fraco}
 - **Tempo** = {Sol, Nublado, Chuva}

Algoritmo ID3 - Árvore de Decisão

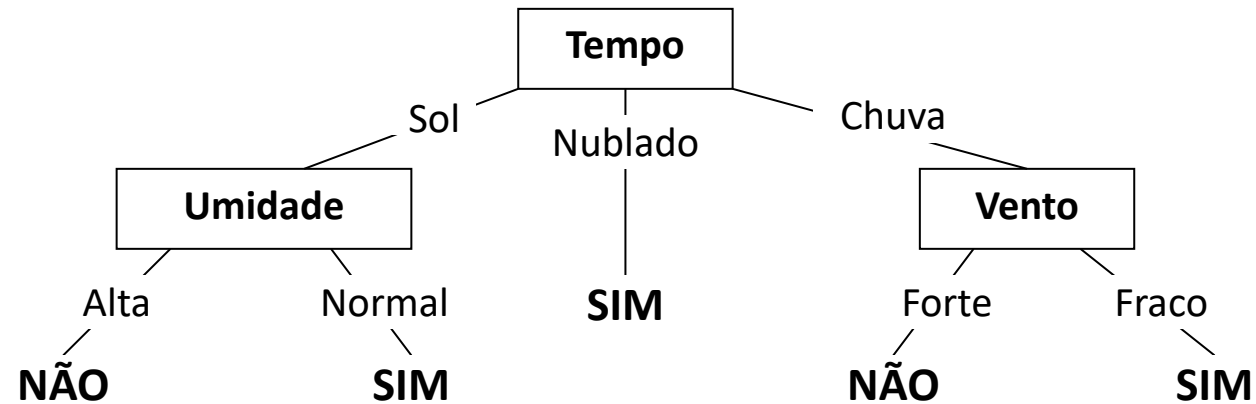
Observando os dias que deveriam ou tiveram partidas de tênis

(conjunto ou exemplos de treinamento):

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Morno	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Morno	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Morno	Normal	Fraco	SIM
11	Sol	Morno	Normal	Forte	SIM
12	Nublado	Morno	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Morno	Alta	Forte	NÃO

Algoritmo ID3 - Árvore de Decisão

- Considerando o conjunto de treinamento, o algoritmo ID3 irá gerar a seguinte árvore de decisão automaticamente:



- Regras: **if** (Tempo=Sol) & (Umidade=Alta) **then** partida_tênis=NÃO;
 if (Tempo=Sol) & (Umidade=Normal) **then** partida_tênis=SIM;
 ...
- Pode-se observar que:
 - A árvore acima é consistente com os exemplos de treinamento

Algoritmo ID3 - Árvore de Decisão

Como construir uma árvore de decisão ?

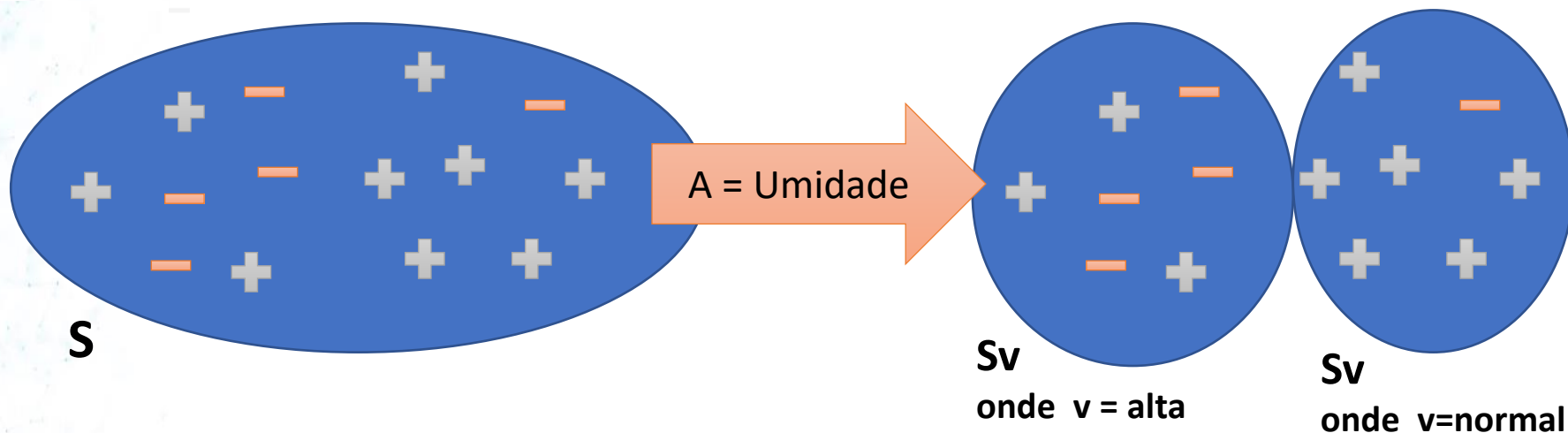
- **Necessita-se:**
 - Ter uma regra que determine qual atributo deve ser considerado em cada nível da árvore.
 - Usar processo de busca para definir a árvore, onde cada estado é uma árvore de decisão parcial (hipótese de uma árvore de decisão)
- O ID3:
 - Usa uma regra chamada **ganho**
 - Usa uma busca gulosa (onde o atributo de maior **ganho** é escolhido) que assemelha-se a busca *Steepest Hill-Climbing*, onde cada estado (nó) é uma árvore de decisão.

Algoritmo ID3 - Árvore de Decisão

Função Ganho de Informação (Gain)

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

Onde **S** é o conjunto de treinamento inicial, **A** é o atributo que irá classificar **S**, e **S_v** é o conjunto de treinamento (sub-conjunto de **S**) quando o atributo **A** tem valor igual a **v**



Algoritmo ID3 - Árvore de Decisão

O que a Função Ganho mede ?

$$Gain(S, A) = Entropia(S) - \sum_{v \in valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

- redução esperada da entropia devido a “classificação” de acordo com o atributo A
- Entropia: quanto mais **puro** for o conjunto, menor é a entropia
 - Conjuntos mais puros: os que possuem só positivos ou só negativos
- *O ganho será maior para o Atributo que dividir melhor o conjunto de treinamento em positivos e negativos.*
 - *A soma das entropias será menor que a entropia original.*

Algoritmo ID3 - Árvore de Decisão

O que é essa tal de Entropia ???

Dado um conjunto de treinamento S , contendo exemplos negativos e positivos, tem a seguinte entropia:

$$\text{Entropia}(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

p_{\oplus} = proporção de ex. positivos em S e p_{\ominus} = proporção de ex. negativos em S

Se a saída não for booleana, então $\text{Entropia}(S) = \sum_{i=0}^n -p_i \log_2 p_i$
para n saídas não-booleanas

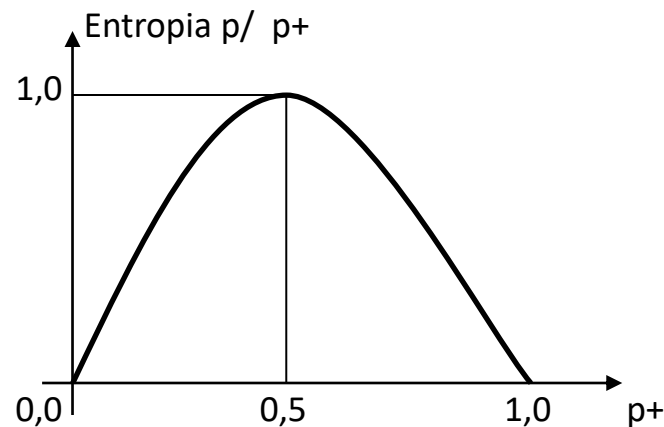
Entropia mede a “impureza” de S !!

Para p_{+} (proporção de positivos)

Entropia = 0 quando tem só positivos

Entropia = 1 quando a proporção é igual

Obs.: $0 \cdot \log_2 0 = 0$



Algoritmo ID3 - Árvore de Decisão

Aplicando a Entropia no Exemplo do jogo de tênis:

S é uma coleção de 14 exemplos da tabela de jogo de tênis, com 9 positivos e 5 negativos

Notação: [9+,5-]

A entropia de S em relação a esta classificação booleana é dada por:

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

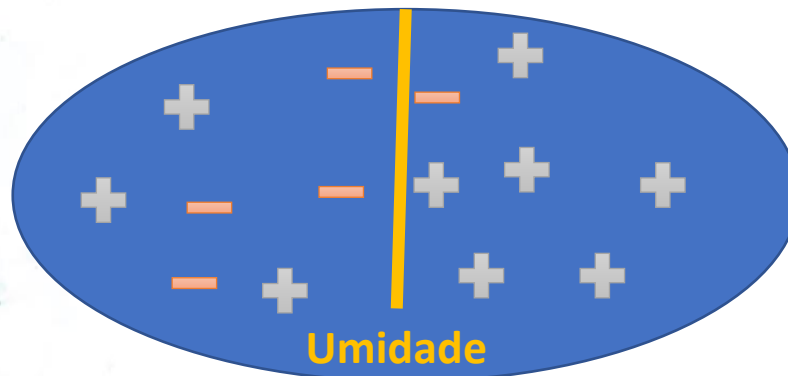
Se considerarmos apenas o atributo TEMPO com valor SOL, teríamos:

$v = \text{Sol}$, $A = \text{TEMPO}$ e S_v com 2+ e 3-

$$Entropia(S_v) = Entropia([2+,3-])$$

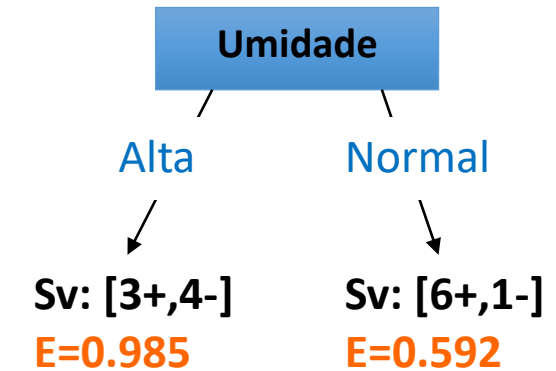
Como gerar uma árvore de decisão ?

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Morno	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Morno	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Morno	Normal	Fraco	SIM
11	Sol	Morno	Normal	Forte	SIM
12	Nublado	Morno	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Morno	Alta	Forte	NÃO



$S = [D1, D2, \dots, D14]$
 $[9+, 5-]$

Entropia = 0.940

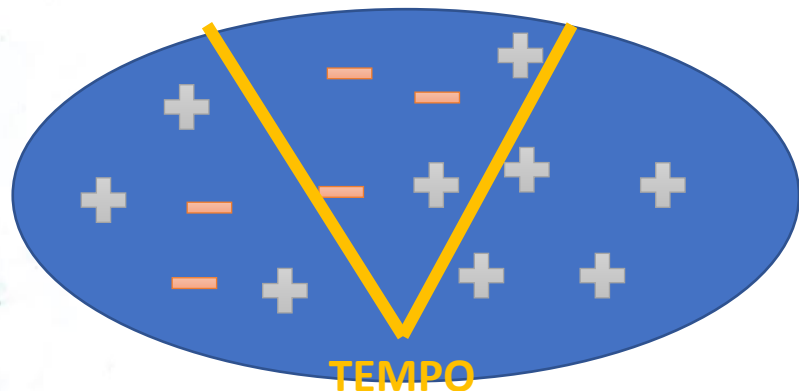


$\text{Ganho}(S, \text{Umidade})$
 $= 0.940 - (7/14) * 0.985 - (7/14) * 0.592$
 $= 0.151$

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

Como gerar uma árvore de decisão ?

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Morno	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Morno	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Morno	Normal	Fraco	SIM
11	Sol	Morno	Normal	Forte	SIM
12	Nublado	Morno	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Morno	Alta	Forte	NÃO



$S = [D1, D2, \dots, D14]$
[9+, 5-]

Entropia = 0.940

Tempo

Sol

Nublado

Chuva

$S_v: [2+, 3-]$

$E=0.970$

$S_v: [4+, 0-]$

$E=0,0$

$S_v: [3+, 2-]$

$E=0.970$

$Ganho(S, TEMPO)$

$$= 0.940 - (5/14) * 0.970 - (4/14) * 0 - (5/14) * 0.970$$

$$= 0.247$$

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

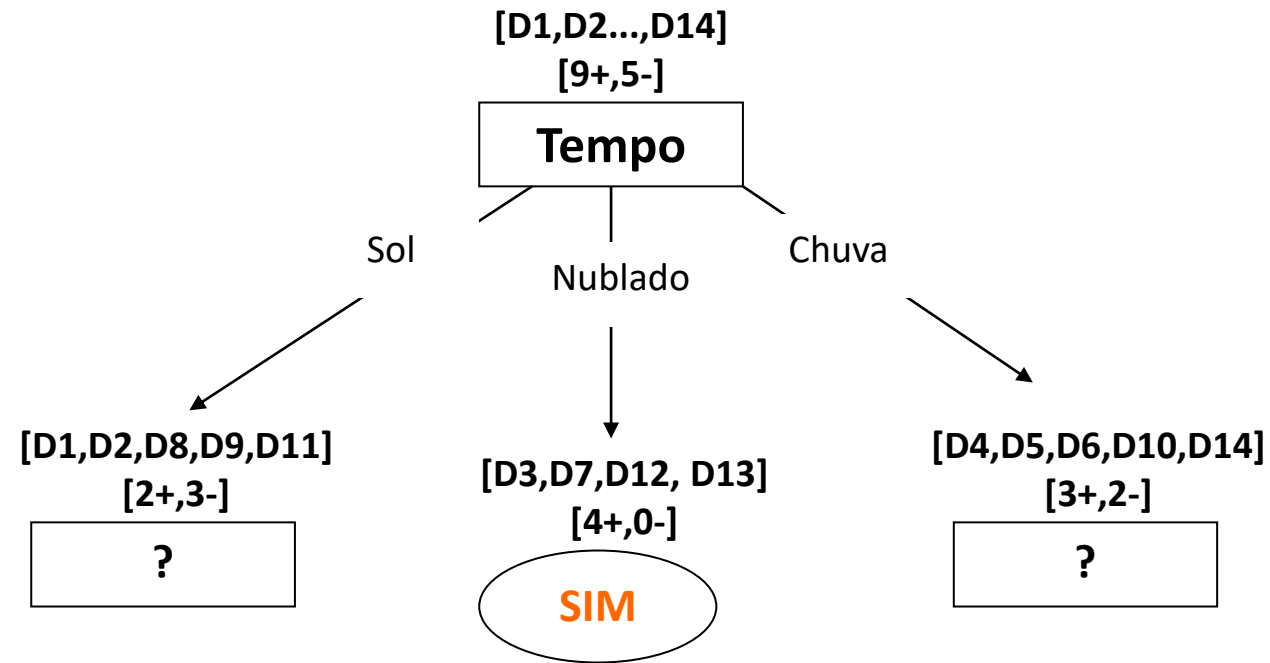
ID3 - Árvore de Decisão

Cálculo do Ganho para os Atributos

- O valor calculado pelo Ganho mostra o quanto cada atributo consegue melhorar a classificação entre positivos e negativos do conjunto original
- Cálculos de todos os ganhos:
 - $\text{Gain}(S, \text{Tempo}) = 0.247$
 - $\text{Gain}(S, \text{Umidade}) = 0.151$
 - $\text{Gain}(S, \text{Vento}) = 0.048$
 - $\text{Gain}(S, \text{Temperatura}) = 0.029$
- Assim, o ID3 escolherá o atributo que oferece maior ganho para cada conjunto de exemplos. Assim, para S, o atributo Tempo será escolhido.

ID3 - Árvore de Decisão

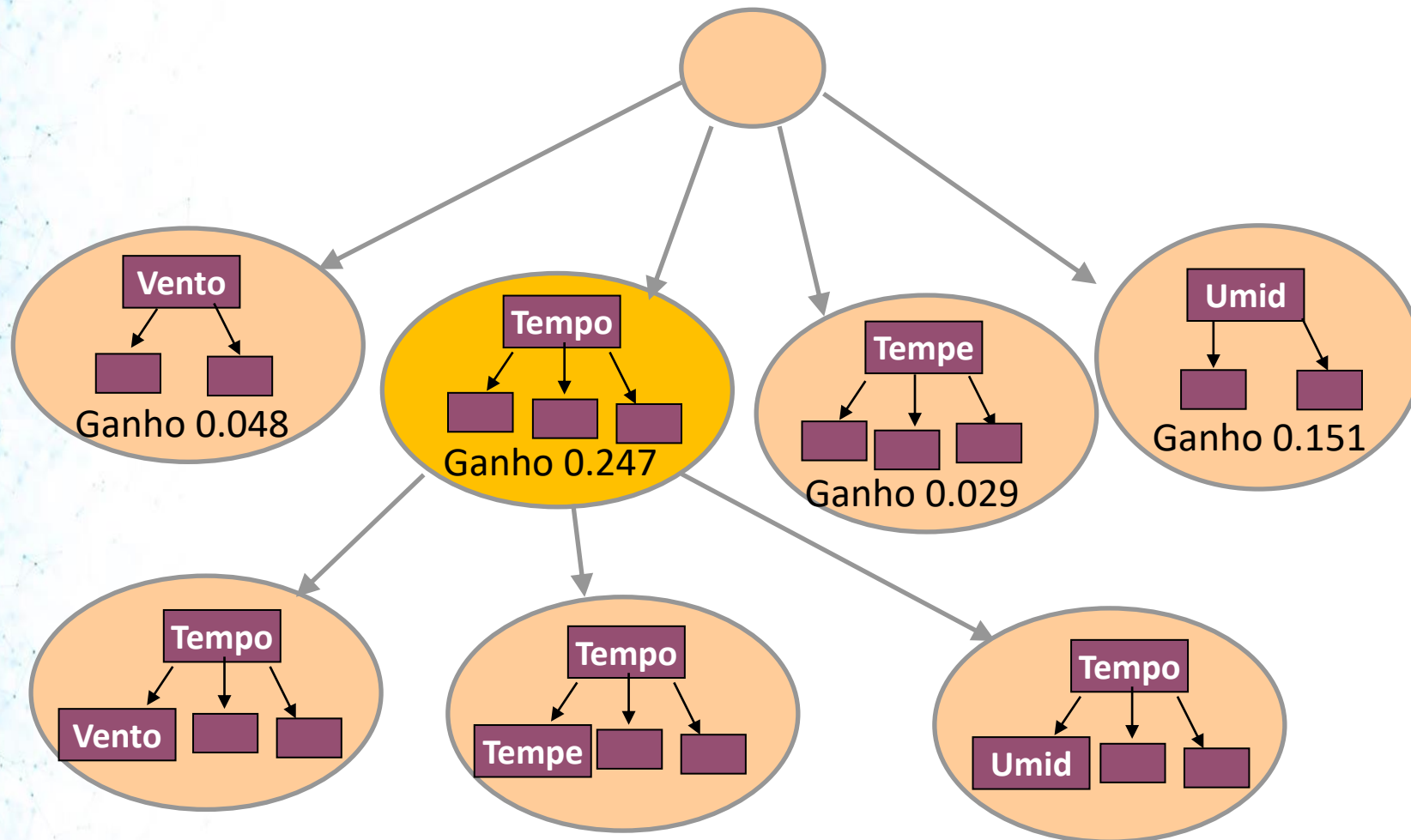
Construindo recursivamente a árvore para a partida de tênis



Em cada '?' o ID3 escolhe qual atributo oferece o melhor ganho com relação aos exemplos já classificados

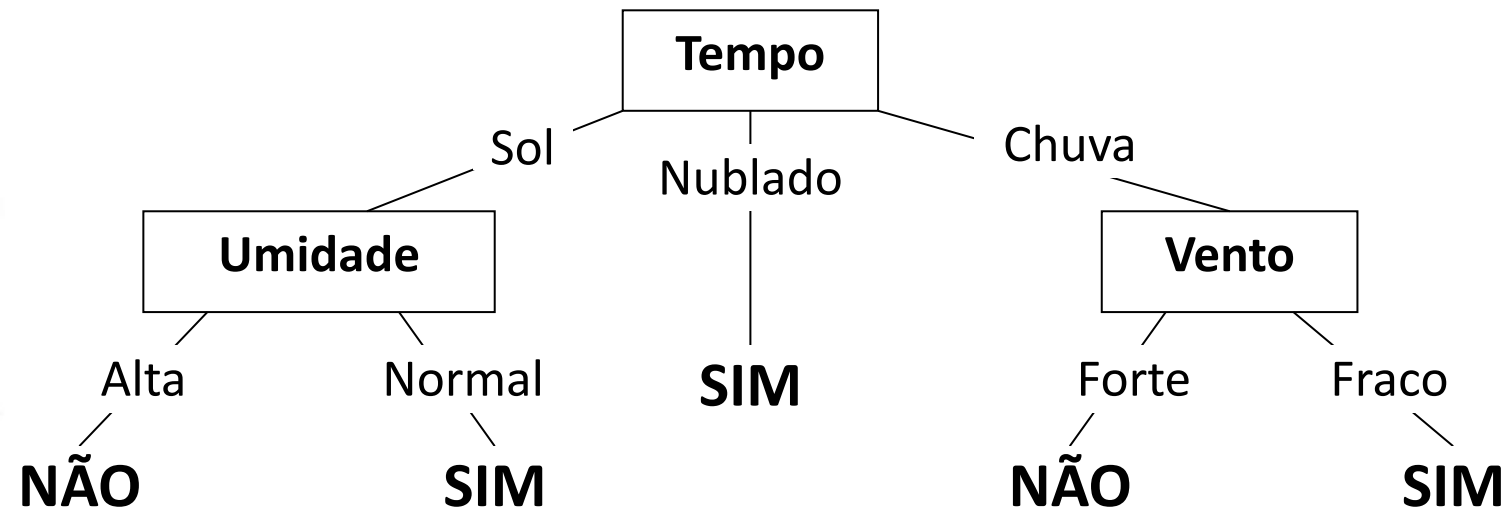
Como achar o resto da árvore ?

- O ID3 usa um sistema de busca tipo *Subida da Encosta pelo caminho + ingreme*, onde a heurística é a função ganho.



ID3 - Árvore de Decisão

- Criando, por fim, a seguinte árvore de decisão:



- Ela é uma boa generalização dos exemplos anteriores ?

ID3 - Árvore de Decisão

- A generalização descrita pela árvore de decisão deve ser consistente com o **conjunto de treinamento**, ou seja, deve satisfazer todas as instâncias, sejam elas positivas ou negativas.
- **Problemas com o conjunto de treinamento (problema da indução):**
- Se **Amostragem for ruim** pode gerar uma árvore de decisão errada.
 - **Ex:** Se a amostragem for feita somente pelos dias 6,7,9,11 e 14, pode-se concluir:
 - Quando Tempo = {Sol ou Nublado} -> há partida de tênis
 - Quando Tempo = {Chuva} -> não há partida de tênis.
 - O que está **errado**, não seria CONSISTENTE com outros possíveis exemplos
- Se **amostragem contiver erro** (chamado ruído) ?
 - Pode-se criar uma árvore errada (NÃO-CONSISTENTE) para outros exemplos além da amostragem
 - Pode-se, inclusive, não conseguir criar a árvore !!!

Como sabemos se aprendizagem funciona ?

- Como sabemos se nossa hipótese (árvore) h está próxima da função desejada f se não conhecemos f ?

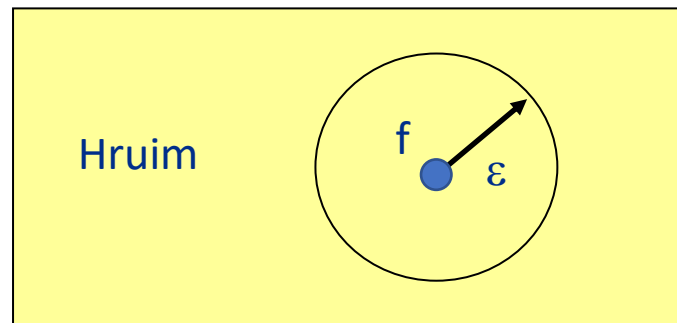
Qualquer hipótese seriamente errada, irá ser desmascarada logo após poucos exemplos. Qualquer hipótese consistente com uma quantidade grande de exemplos, terá pouca probabilidade de estar seriamente errada.

- A hipótese h será **provavelmente aproximadamente correta:**
aprendizagem PAC

Quantos exemplos precisamos ter ?

- Vamos fazer uma análise. Consideraremos:
 - f a função verdadeira (a árvore ótima)
 - H o conjunto de hipóteses (árvores) possíveis
 - N o conjunto de exemplos de treinamento
- A função f , é um elemento de H .
- Uma hipótese h é dita aproximadamente correta se $\text{erro}(h)$ com relação a f seja menor que um ϵ

H



Quantidade de exemplos necessários

- Mostra-se que:

$$N \geq \frac{1}{\varepsilon} \left(\ln \frac{1}{\delta} + \ln |H| \right)$$

- Onde δ é um número pequeno.
- Conseguiremos, portanto, uma hipótese h abaixo do erro ε se o algoritmo ver N exemplos de treinamento.
- Mas como estimar H ?
 - H para n atributos booleanos: 2^{2^n}
 - Mas H é muito grande... Teríamos que ler um número muito grande de N !!!
- Para escapar disso: mesmo lendo menos exemplos que o necessário, manteremos hipóteses mais simples (e genéricas)
Occam's razor = prefira a hipótese mais simples possível

Erro da Hipótese

- Erro e precisão
 - Taxa de erro de uma árvore

$$err(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{y_i \neq h(x_i)}$$

- **Precisão:** $acc(h) = 1 - err(h)$

Onde:

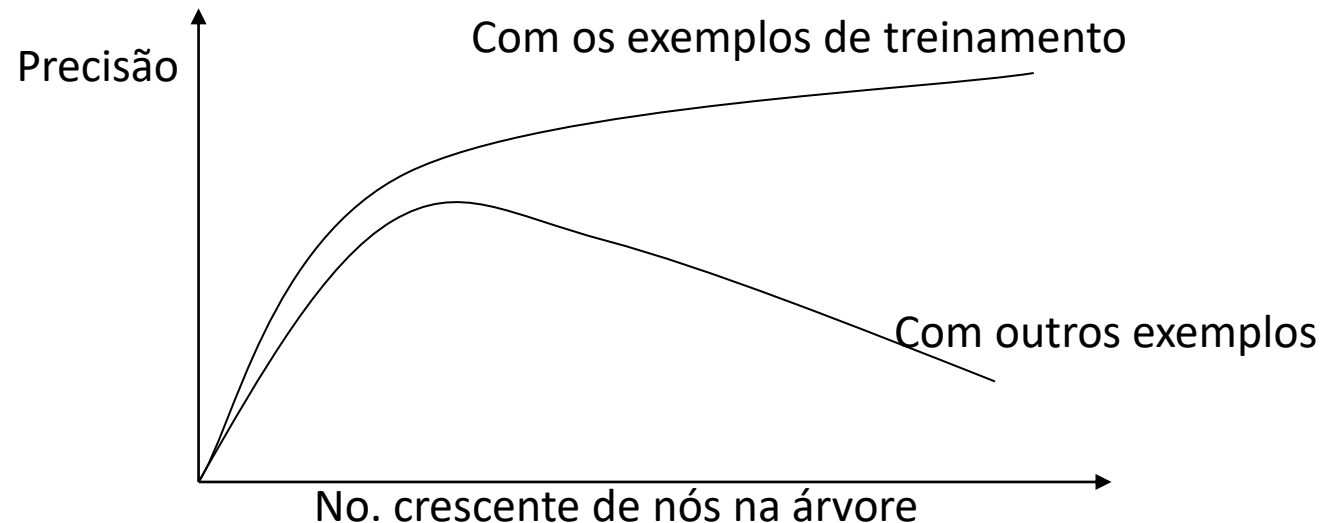
- h = classificador
- n = numero de exemplos
- y_i é a saída correta do exemplo x_i
- $\mathbb{I}_{E} = 1$ se $y_i = h(x_i)$
0 se $y_i \neq h(x_i)$

Para cada Classe $C_j \in T$

Overfitting – Hiper-especialização

Problema de todos algoritmos de aprendizagem!!

- **Definição:** dado um espaço de hipóteses H , uma hipótese $h \in H$ **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.



- Como solucionar o problema: deixar a árvore o mais simples possível (Occam's razor)

Solucionando o Overfitting

1a. Solução:

Parar de crescer a árvore antes de alcançar o ponto de classificação perfeita dos exemplos de treinamento. Mas, quando parar?

- **Validação cruzada:** tenta estimar quão bem a hipótese corrente irá prever dados ainda não recebidos (“vistos”). Segue os seguintes passos:
 - 1. Divide-se o conjunto de treinamento em dois conjuntos: Conjunto de Treinamento (CT) e Conjunto de Validação (CV)
 - 2. Cria-se a árvore de decisão a partir de CT e usa CV para verificar a percentagem de erro.
 - Repete-se os passos 1 e 2 para diversos CT e CV’s diferentes e aleatórios
- Com isso, determina-se a melhor árvore para ser usada.

Solucionando Overfitting

2a. Solução

Abordagens que provoquem o *overfitting* e depois podam a árvore (*post-pruning*)

Método do Erro Reduzido:

- Considera-se cada nó como candidato à folha da árvore (elimina sub-árvore abaixo dele);
- o nó se torna folha (nova árvore) sempre que a precisão da classificação não diminuir em relação à árvore original
- Usa-se os exemplos de treinamento para inferir uma amostragem estatística, extraíndo intervalo de confiança e variância. Use esta amostragem para definir taxas de erros para ramos e sub-ramos da árvore.

Além do ID3...

O ID3 tem alguns problemas e limitações:

- O atributo com mais valores tende a obter o maior ganho
 - Isso causa problemas, pois nem sempre este é o melhor atributo de separação.
 - E.g.: Imagine Data=12.7.1996 como atributo distribuídos em sub-conjuntos de 1 dia ?

Além desse problema, o ID3 tem algumas limitações

- Não trabalha com atributos numéricos
- Não aceita informação incompleta
- Não trabalha com atributos com custos



Melhorando o ID3.....

Regra de Ganho

$$\textit{Gain} (S, A) = \textit{Entropia} (S) - \sum_{v \in \textit{valores}(A)} \frac{|S_v|}{|S|} \textit{Entropia}(S_v)$$

Para resolver a tendência de um Atributo A ter o maior ganho apenas porque possui mais valores, a regra ganho pode ser normalizada pela quantidade de valores de um atributo (**Razão de Ganho**):

$$\textit{Gain} (S, A) = \frac{\textit{Entropia} (S) - \sum_{v \in \textit{valores}(A)} \frac{|S_v|}{|S|} \textit{Entropia}(S_v)}{- \sum_{v \in \textit{valores}(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}}$$

Atributos com custos

Por exemplo:

- Diagnóstico Médico: teste de sangue custa 1000 reais

Como aprender uma árvore consistente com baixo custo estimado ?

Propostas de modificação da regra Ganho:

- $\text{Gain}^2(S,A)/\text{Cost}(A)$ [Tan, Schimmer 1990]

Atributos Numéricos

- ❑ Método Padrão: Separação Binária (Ex: $\text{temp} < 45$)
- ❑ Todo atributo numérico tem diversos pontos de separação.
- ❑ Solução:
 - Calcule o ganho para cada ponto de separação
 - Escolha o o melhor ponto de separação (o que dá maior ganho)
 - O ganho desse melhor ponto será o valor de ganho do atributo

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- Ex: $\text{temperatura} < 71.5$: yes/4, no/2
 $\text{temperatura} \geq 71.5$: yes/5, no/3

- ❑ **Computacionalmente muito caro !!!!**
 - ❑ Testa todas as possibilidades de separação para escolher o maior ganho

Atributos Numéricos

- Podemos calcular os pontos de separação mais rapidamente:

Valor	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Classe	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- Os pontos de separação só precisam ser calculados nos pontos entre classes diferentes (Fayyad & Irani, 1992)
- Pontos intermediários às pontos de separação acima nunca serão pontos de separação ótimos (**nunca terão o maior ganho**)

Informação Incompleta

... E se alguns valores de atributos estão faltando ?

- 1a solução
 - Usando os exemplos de treinamento
 - Se o nó n testa o atributo A , atribua ao valor que falta o valor mais comum de A entre todos os exemplos escolhidos para o nó n .
 - **Por exemplo, MORNO para o atributo TEMPERATURA no ex. Jogo de tênis. São 6 em 14 exemplos.**

Dia	Tempo	Temperatura	Unidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva		Alta	Forte	NÃO

Dia	Tempo	Temperatura	Unidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Momo	Alta	Forte	NÃO

Informação Incompleta

2a. Solução possível

- Também com o uso dos exemplos de treinamento
- Atribua ao valor que está faltando o valor mais comum de A usados em outros exemplos que tenham o mesmo resultado.
 - **Valor ALTA para o atributo UMIDADE para as partidas em que NÃO ocorreram partida de tênis.**

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Momo		Forte	NÃO

Dia	Tempo	Temperatura	Umidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva	Momo	Alta	Forte	NÃO

Informação Incompleta

3a. Solução possível

- Também com o uso dos exemplos de treinamento
- Atribua probabilidade p_i para cada possível valor de A. Use uma fração dos valores de A conforme p_i ao avançar na árvore.
- Em um valor de temperatura faltando, considere:
 - 0,36 para SOL, 0,36 para CHUVA e 0,28 para NUBLADO



Dia	Tempo	Temperatura	Unidade	Vento	Partida
1	Sol	Quente	Alta	Fraco	NÃO
2	Sol	Quente	Alta	Forte	NÃO
3	Nublado	Quente	Alta	Fraco	SIM
4	Chuva	Momo	Alta	Fraco	SIM
5	Chuva	Frio	Normal	Fraco	SIM
6	Chuva	Frio	Normal	Forte	NÃO
7	Nublado	Frio	Normal	Forte	SIM
8	Sol	Momo	Alta	Fraco	NÃO
9	Sol	Frio	Normal	Fraco	SIM
10	Chuva	Momo	Normal	Fraco	SIM
11	Sol	Momo	Normal	Forte	SIM
12	Nublado	Momo	Alta	Forte	SIM
13	Nublado	Quente	Normal	Fraco	SIM
14	Chuva		Alta	Forte	NÃO

C4.5 e C5.0

- **C4.5 é uma extensão do ID3.**
- **Algumas características adicionais:**
 - Trabalha com atributos com números contínuos
 - Resolve o problema da regra de Ganho usando Razão_de_Ganho
 - Pós-poda baseada no conjunto de treinamento visando a diminuição do *overfitting* e conseqüente aumento da precisão.
 - C4.5 pode lidar com informação incompleta (falta de algum atributo, por exemplo)
- Versão C5.0 comercial: mais precisa e mais veloz (usa outras técnicas mais modernas)

Quinlan, J. Ross (ID3 e C4.5)



Aplicações de Aprendizado Indutivo

- Automação Residencial Inteligente
 - Sistema aprende regras com o Habitante
 - Aprende comportamentos
- Controle de SPAM
 - Aprende automaticamente as características de SPAMs
 - Valida constantemente as regras anti-SPAM
- Automóvel Inteligente
 - Aprende o modo de dirigir do motorista

Bibliografia desta aula

aprofundamento nos assuntos desta aula, segue a seguinte referência bibliográfica

- Rich, E. (Inteligência Artificial)
 - Capítulo 17 (Aprendizado)
- Russel & Norvig (Artificial Intelligence)
 - Capítulo 18 (Aprendizado)
- Rezende, Solange (Sistemas Inteligentes)
 - Caps. 4 e 5 (Aprendizado)
- Alguns slides desta aula foram baseados no slides:
- Anna Realí Costa: “Aprendizado: ID3”, Poli-USP.
- Slides Machine Learning 2D5362 – Lecture 3 – Decision Tree Learning
- Joost N. Kok: “Machine Learning in Real World: C4.5”, Leiden University – Holanda