



Indução de Árvores de Decisão

- Várias aplicações em Inteligência Artificial em tarefas de importância prática são baseadas na construção de um modelo de conhecimento que é utilizado por um especialista humano
- O objetivo desta aula é fornecer conceitos básicos sobre indução de árvores de decisão



Histórico

- 1960's
 - 1966: Hunt e colegas em psicologia usaram métodos de busca exaustiva em árvores de decisão para modelar o aprendizado de conceitos humanos
- 1970's
 - 1977: Breiman, Friedman, e colegas em estatística desenvolveram *Classification And Regression Trees* (CART)
 - 1979: Primeiro trabalho de Quinlan com proto-ID3 (*Induction of Decision Trees*)
- 1980's
 - 1984: primeira publicação em massa do software CART (presente atualmente em vários produtos comerciais)
 - 1986: Artigo de Quinlan sobre ID3
 - Variedade de melhorias: tratamento de ruído, atributos contínuos, atributos com valores desconhecidos, árvores obliquas (não paralelas aos eixos), etc
- 1990's
 - 1993: Algoritmo atualizado de Quinlan: C4.5 (release 8)
 - Maior poder, heurísticas de controle de *overfitting* (C5.0, etc.); combinando DTs

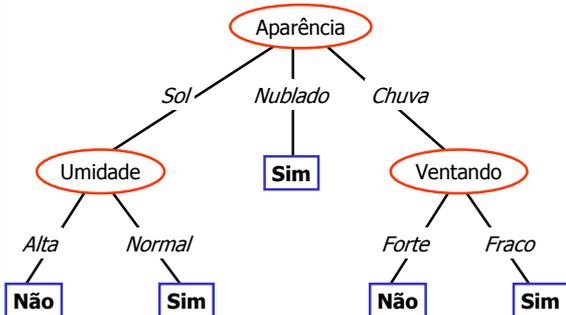
TDIDT

- Os algoritmos de classificação cujo conhecimento adquirido é representado como Árvore de Decisão (DT) pertencem a família TDIDT (*Top Down Induction of Decision Trees*)
- Árvore de Decisão: estrutura recursiva definida como:
 - um nó folha que indica uma classe, ou
 - um nó de decisão contém um teste sobre o valor de um atributo. Cada resultado do teste leva a uma sub-árvore. Cada sub-árvore tem a mesma estrutura da árvore

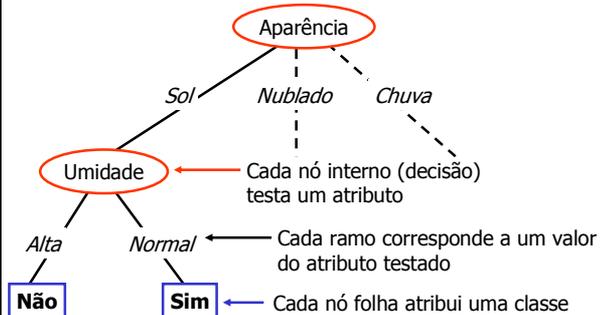
DT para Jogar Tênis

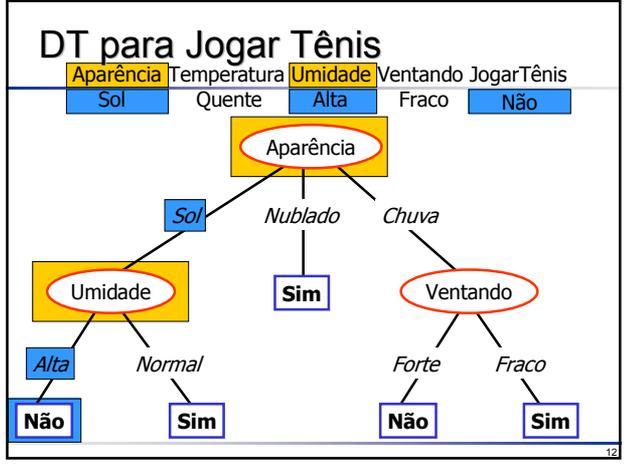
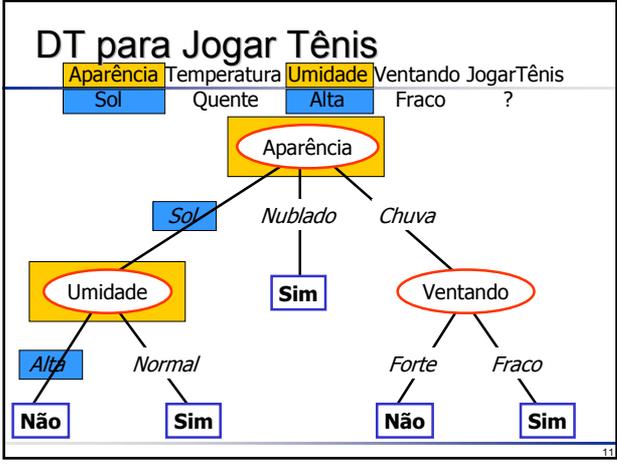
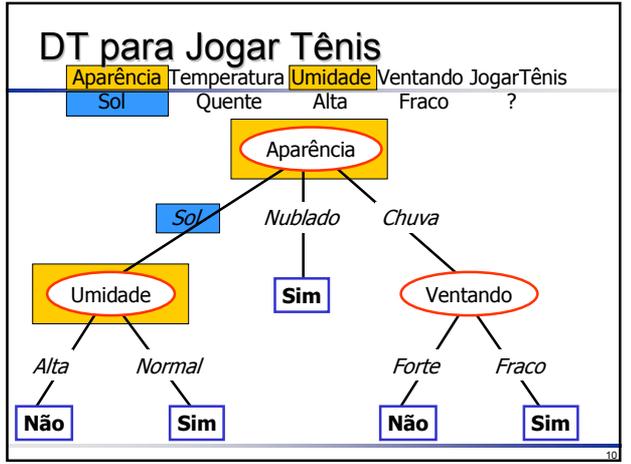
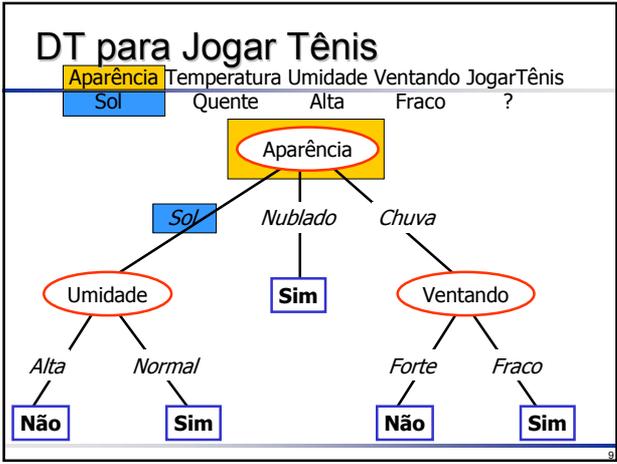
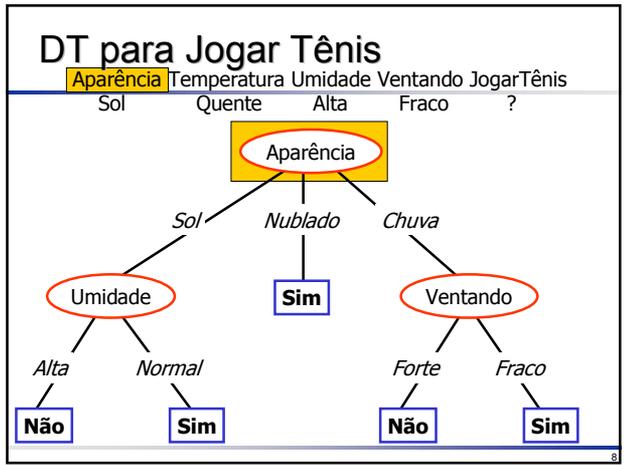
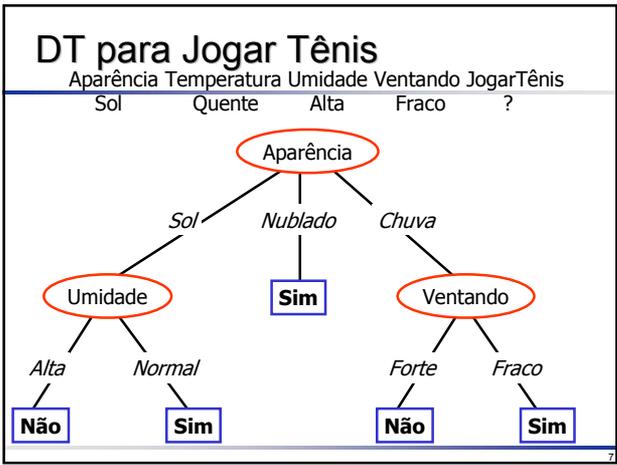
- Atributos:
 - Aparência: *Sol, Nublado, Chuva*
 - Umidade: *Alta, Normal*
 - Ventando: *Forte, Fraco*
 - Temperatura: *Quente, Média, Fria*
 - Classe (Conceito Alvo) – jogar tênis: *Sim, Não*

DT para Jogar Tênis

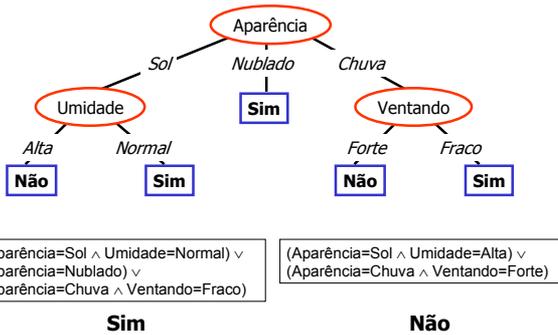


DT para Jogar Tênis



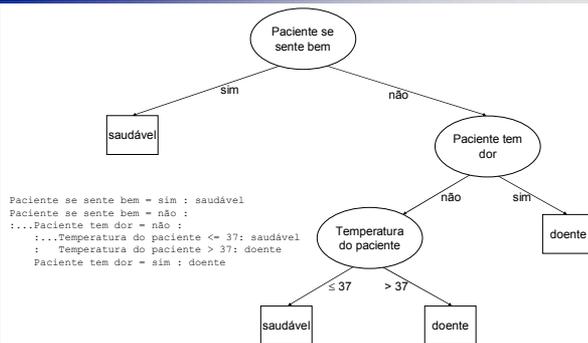


DTs Representam Disjunções de Conjunções



13

Exemplo: Árvore de Decisão



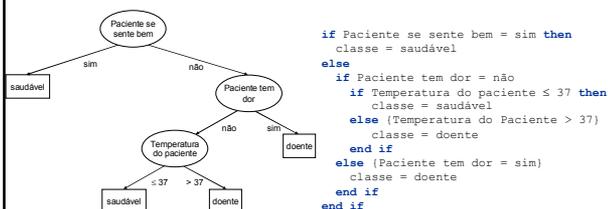
14

Representação da DT como um Conjunto de Regras

- Uma árvore pode ser representada como um conjunto de regras
- Cada regra começa na raiz da árvore e caminha para baixo, em direção às folhas
 - Cada nó de decisão acrescenta um teste às premissas (condições) da regra
 - O nó folha representa a conclusão da regra

15

Representação da DT como um Conjunto de Regras



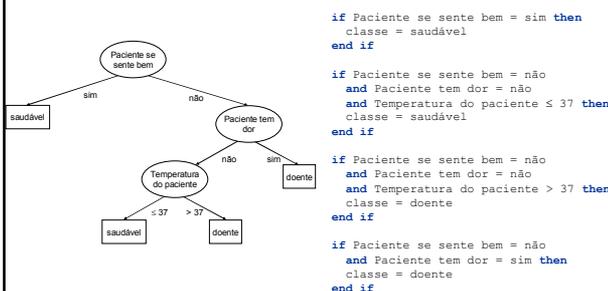
16

Representação da DT como um Conjunto de Regras Disjuntas

- As regras representadas por uma árvore de decisão são disjuntas
- Assim, elas podem ser escritas como regras **separadas**, começando pela raiz, e, conseqüentemente, o *else* não é necessário

17

Representação da DT como um Conjunto de Regras Disjuntas



18

Algoritmo TDIDT

- ❑ Se todos os exemplos de treinamento pertencem a uma única classe então a árvore é uma folha rotulada com a respectiva classe
- ❑ Caso contrário:
 - selecione um teste baseado em um atributo
 - divida o conjunto de treinamento em subconjuntos, cada um correspondendo a um dos possíveis (mutuamente exclusivos) valores para o teste
 - aplique o mesmo processo para cada subconjunto

Algoritmo TDIDT

- ❑ Seja T um conjunto de exemplos de treinamento com classes $\{C_1, C_2, \dots, C_k\}$. Há três possibilidades:
 - 1) T contém um ou mais exemplos, todos pertencendo a uma mesma classe C_i ; a árvore de decisão para T é uma folha identificando a classe C_i
 - 2) T não contém exemplos: a árvore de decisão é novamente uma folha, mas a classe associada com a folha deve ser determinada por alguma informação além de T . Por exemplo, a folha pode ser escolhida de acordo com algum conhecimento do domínio, tal como a classe majoritária. C4.5 utiliza a classe mais frequente do nó pai deste nó (folha)
 - 3) T contém exemplos que pertencem a uma mistura de classes: nesta situação a idéia é refinar T em subconjuntos que são (ou aparentam ser) coleções de exemplos de uma única classe. Um teste é escolhido, baseado em um único atributo, com resultados mutuamente exclusivos. Sejam os possíveis resultados do teste denotados por $\{O_1, O_2, \dots, O_r\}$. T é então particionado em subconjuntos T_1, T_2, \dots, T_r , nos quais cada T_i contém todos os exemplos em T que possuem como resultado daquele teste o valor O_i . A árvore de decisão para T consiste em um nó (interno) identificado pelo teste escolhido e uma aresta para cada um dos resultados possíveis. Para cada partição, pode-se exigir que cada T_i contenha um número mínimo de exemplos, evitando partições com poucos exemplos. O default de C4.5 é de 2 exemplos
- ❑ Os passos 1, 2 e 3 são aplicados recursivamente para cada subconjunto de exemplos de treinamento de forma que, em cada nó, as arestas levam para as sub-árvores construídas a partir do subconjunto de exemplos T_i
- ❑ Após a construção da árvore de decisão, a poda pode ser realizada para melhorar sua capacidade de generalização

Classificando Novos Exemplos

- ❑ Uma DT pode ser usada para classificar novos exemplos (nunca vistos)
- ❑ A partir da raiz basta descer através dos nós de decisão até encontrar um nó folha: a classe correspondente a esse nó folha é a classe do novo exemplo
- ❑ Um exemplo (sem valores desconhecidos) é classificado apenas por uma regra (sub-árvore)

Exemplo (adaptado de Quinlan, 93)

- ❑ Neste exemplo, vamos considerar um conjunto de exemplos que contém medições diárias sobre condições meteorológicas
- ❑ Atributos
 - aparência: “sol”, “nublado” ou “chuva”
 - temperatura: temperatura em graus Celsius
 - umidade: umidade relativa do ar
 - ventando: “sim” ou “não”
- ❑ Cada exemplo foi rotulado com “bom” se nas condições meteorológicas daquele dia é aconselhável fazer uma viagem à fazenda e “ruim”, caso contrário

O Conjunto de Dados “Viagem”

Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom

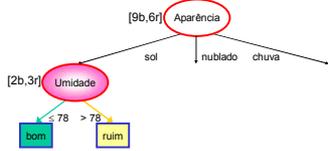
Escolhendo “Aparência” para Particionar

Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



Escolhendo "Umidade" para Particionar "Aparência=sol"

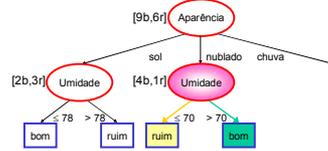
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



26

Escolhendo "Umidade" para Particionar "Aparência=nublado"

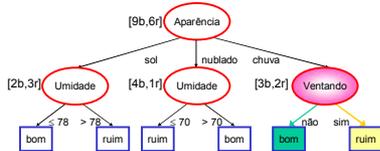
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



27

Escolhendo "Ventando" para Particionar "Aparência=chuva"

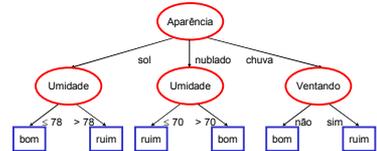
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



28

Árvore de Decisão Induzida (sem poda)

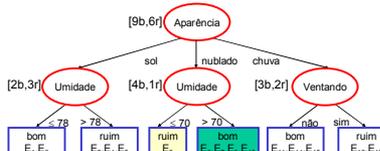
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



29

Árvore de Decisão Induzida (sem poda)

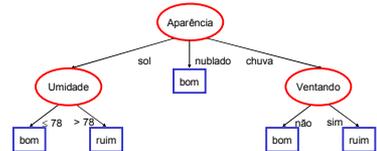
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



30

Árvore de Decisão Induzida (podada)

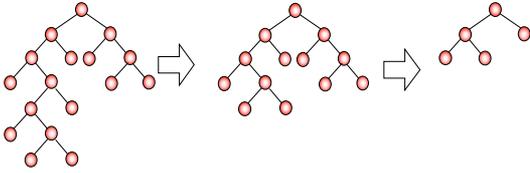
Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar
E ₁	sol	25	72	sim	bom
E ₂	sol	28	91	sim	ruim
E ₃	sol	22	70	não	bom
E ₄	sol	23	95	não	ruim
E ₅	sol	30	85	não	ruim
E ₆	nublado	23	90	sim	bom
E ₇	nublado	29	78	não	bom
E ₈	nublado	19	65	sim	ruim
E ₉	nublado	26	75	não	bom
E ₁₀	nublado	20	87	sim	bom
E ₁₁	chuva	22	95	não	bom
E ₁₂	chuva	19	70	sim	ruim
E ₁₃	chuva	23	80	sim	ruim
E ₁₄	chuva	25	81	não	bom
E ₁₅	chuva	21	80	não	bom



31

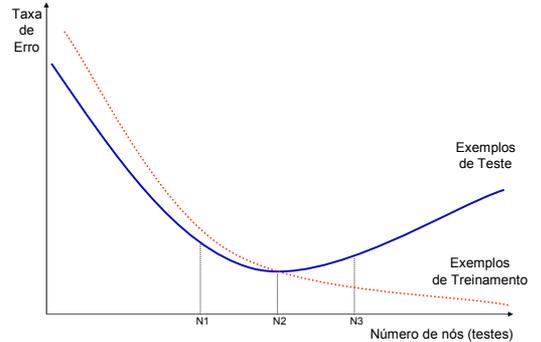
(Pós-)Poda

- Uma árvore maior é induzida de forma a superajustar os exemplos e então ela é podada até obter uma árvore menor (mais simples)
- A poda evita *overfitting*



32

Relação entre Tamanho da Árvore de Decisão e a Taxa de Erro



33

Escolha do Atributo

- A maioria dos algoritmos de construção de árvores de decisão são sem retrocesso (sem *backtracking*) ou seja, gulosos (*greedy*)
- Uma vez que um teste foi selecionado para particionar o conjunto atual de exemplos, a escolha é fixada e escolhas alternativas não são exploradas

34

Escolha do Atributo

- A chave para o sucesso de um algoritmo de aprendizado por árvores de decisão depende do critério utilizado para escolher o atributo que particiona o conjunto de exemplos em cada iteração
- Algumas possibilidades para escolher esse atributo são:
 - aleatória: seleciona qualquer atributo aleatoriamente
 - menos valores: seleciona o atributo com a menor quantidade de valores possíveis
 - mais valores: seleciona o atributo com a maior quantidade de valores possíveis
 - ganho máximo: seleciona o atributo que possui o maior ganho de informação esperado, isto é, seleciona o atributo que resultará no menor tamanho esperado das subárvores, assumindo que a raiz é o nó atual;
 - razão de ganho
 - índice Gini

35

Entropia

- Seja S um subconjunto de T
- A informação esperada (ou entropia) do subconjunto S é (em bits) dado por

$$\text{info}(S) = - \sum_{j=1}^k p(C_j, S) \times \log_2(p(C_j, S))$$

$$p(C_j, S) = \frac{\text{freq}(C_j, S)}{|S|} = \frac{\text{número de exemplos em } S \text{ com classe } C_j}{\text{número de exemplos em } S}$$

- Quando aplicado a todo o conjunto de treinamento T , **info(T)** mede a quantidade média de informação necessária para identificar a classe de um exemplo em T
- Lembrar que $\log_b(a) = \ln(a) / \ln(b)$, ou seja, $\log_2(x) = \ln(x) / \ln(2)$
- Observação: assumir $0 \cdot \log_2(0) = 0$

36

Exercício

- Calcule $\text{info}(T)$ para
 - Um conjunto T de 64 exemplos, sendo 29 exemplos da classe positiva e 35 da classe negativa, ou seja, $[29+, 35-]$
 - Um conjunto T de 64 exemplos, sendo 20 exemplos da classe positiva, 32 da classe negativa e 12 da classe asterisco, ou seja, $[20+, 32-, 12*]$
 - Idem para $T = [20+, 32-, 6*, 6\$]$

37

Solução

- $T = [29+, 35-]$
 - $\text{info}(T) = \text{info}([29+, 35-])$
 $= -29/64 \log_2 29/64 - 35/64 \log_2 35/64$
 $= 0.99$
- $T = [20+, 32-, 12^*]$
 - $\text{info}(T) = \text{info}([20+, 32-, 12^*])$
 $= -20/64 \log_2 20/64 - 32/64 \log_2 32/64$
 $- 12/64 \log_2 12/64$
 $= 1.48$
- $T = [20+, 32-, 6^*, 6\$]$
 - $\text{info}(T) = \text{info}([20+, 32-, 6^*, 6\$])$
 $= -20/64 \log_2 20/64 - 32/64 \log_2 32/64$
 $- 6/64 \log_2 6/64 - 6/64 \log_2 6/64$
 $= 1.66$

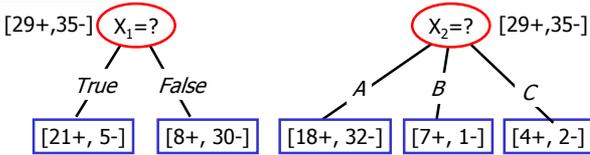
Entropia

- Considere agora que T foi particionado de acordo com r valores do atributo X , ou seja $X = O_1, X = O_2, \dots, X = O_r$, gerando os subconjuntos T_1, T_2, \dots, T_r , respectivamente
 - T_i é o formado pelos exemplos de T nos quais o atributo $X = O_i$, ou seja, $T_i = \{vz \in T: X = O_i\}$
- A informação esperada para este particionamento é a soma ponderada sobre todos os subconjuntos T_i :

$$\text{info}(X, T) = \sum_{i=1}^r \frac{|T_i|}{|T|} \times \text{info}(T_i)$$

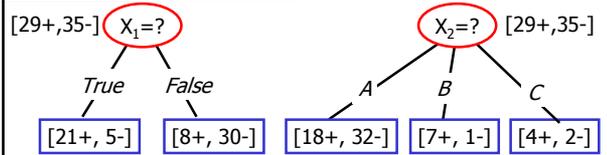
- lembrando que $|T|$ é a cardinalidade do conjunto T

Exercício



Calcule $\text{info}(X_1, T)$ e $\text{info}(X_2, T)$, $T = [29+, 35-]$

Solução



$$\begin{aligned} \text{info}([21+, 5-]) &= 0.71 \\ \text{info}([8+, 30-]) &= 0.74 \\ \text{info}(X_1, [29+, 35-]) &= \\ &= -26/64 * \text{info}([21+, 5-]) \\ &= -38/64 * \text{info}([8+, 30-]) \\ &= 0.73 \end{aligned}$$

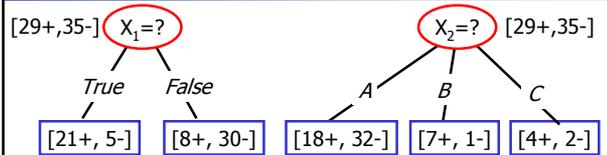
$$\begin{aligned} \text{info}([18+, 32-]) &= 0.94 \\ \text{info}([7+, 1-]) &= 0.54 \\ \text{info}([4+, 2-]) &= 0.92 \\ \text{info}(X_2, [29+, 35-]) &= -50/64 * \text{info}([18+, 32-]) - \\ &= 8/64 * \text{info}([7+, 1-]) - 6/64 * \text{info}([4+, 2-]) \\ &= 0.89 \end{aligned}$$

Ganho de Informação

- A quantidade
 - $\text{gain}(X, T) = \text{info}(T) - \text{info}(X, T)$
 - mede o ganho de informação pela partição de T de acordo com o atributo X
- O critério de ganho (ganho máximo) seleciona o atributo $X \in T$ (ou seja, $X \in \{X_1, X_2, \dots, X_m\}$) que maximiza o ganho de informação

$$\max_{X \in \{X_1, X_2, \dots, X_m\}} \text{gain}(X, T)$$

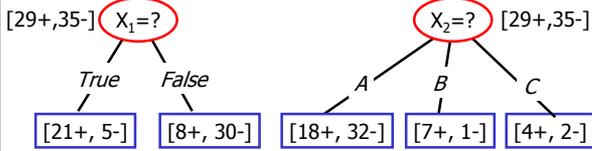
Exercício



$$\begin{aligned} T &= [29+, 35-] \\ \text{info}([29+, 35-]) &= 0.99 \\ \text{info}(X_1, [29+, 35-]) &= 0.73 \\ \text{info}(X_2, [29+, 35-]) &= 0.89 \end{aligned}$$

Qual o ganho de X_1 ? E de X_2 ?
Com qual atributo obtém-se o ganho máximo?

Solução



$T = [29+, 35-]$

$\text{info}([29+, 35-]) = 0.99$

$\text{info}(X_1, [29+, 35-]) = 0.73$

$\text{info}(X_2, [29+, 35-]) = 0.89$

$\text{gain}(X_1, T) = \text{info}(T) - \text{info}(X_1, T)$

$= 0.99 - 0.73 = \mathbf{0.26}$

$\text{gain}(X_2, T) = \text{info}(T) - \text{info}(X_2, T)$

$= 0.99 - 0.89 = 0.10$

Ganho máximo é obtido com X_1

Exemplo

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	quente	alta	falso	não
E ₂	sol	quente	alta	verdadeiro	não
E ₃	nublado	quente	alta	falso	sim
E ₄	chuva	agradável	alta	falso	sim
E ₅	chuva	fria	normal	falso	sim
E ₆	chuva	fria	normal	verdadeiro	não
E ₇	nublado	fria	normal	verdadeiro	sim
E ₈	sol	agradável	alta	falso	não
E ₉	sol	fria	normal	falso	sim
E ₁₀	chuva	agradável	normal	falso	sim
E ₁₁	sol	agradável	normal	verdadeiro	sim
E ₁₂	nublado	agradável	alta	verdadeiro	sim
E ₁₃	nublado	quente	normal	falso	sim
E ₁₄	chuva	agradável	alta	verdadeiro	não

Exemplo

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	quente	alta	falso	não
E ₂	sol	quente	alta	verdadeiro	não
E ₃	nublado	quente	alta	falso	sim
E ₄	chuva	agradável	alta	falso	sim
E ₅	chuva	fria	normal	falso	sim
E ₆	chuva	fria	normal	verdadeiro	não
E ₇	nublado	fria	normal	verdadeiro	sim
E ₈	sol	agradável	alta	falso	não
E ₉	sol	fria	normal	falso	sim
E ₁₀	chuva	agradável	normal	falso	sim
E ₁₁	sol	agradável	normal	verdadeiro	sim
E ₁₂	nublado	agradável	alta	verdadeiro	sim
E ₁₃	nublado	quente	normal	falso	sim
E ₁₄	chuva	agradável	alta	verdadeiro	não

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\text{info}(T) = -\sum_{j=1}^n p(C_j, T) \times \log_2(p(C_j, T))$$

$$= -p(\text{sim}, T) \times \log_2(p(\text{sim}, T)) - p(\text{não}, T) \times \log_2(p(\text{não}, T))$$

$$= -\frac{9}{14} \times \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \times \log_2\left(\frac{5}{14}\right)$$

$$= 0.94029 \text{ bits}$$

$$\text{info}(\text{Aparência}, T) = \sum_{i=1}^3 \frac{|T_i|}{|T|} \times \text{info}(T_i)$$

$$= \frac{|T_{\text{sol}}|}{|T|} \times \text{info}(T_{\text{sol}}) + \frac{|T_{\text{nublado}}|}{|T|} \times \text{info}(T_{\text{nublado}}) + \frac{|T_{\text{chuva}}|}{|T|} \times \text{info}(T_{\text{chuva}})$$

$$= \frac{5}{14} \times \text{info}(T_{\text{sol}}) + \frac{4}{14} \times \text{info}(T_{\text{nublado}}) + \frac{5}{14} \times \text{info}(T_{\text{chuva}})$$

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\text{info}(T_{\text{sol}}) = -\sum_{j=1}^n p(C_j, T_{\text{sol}}) \times \log_2(p(C_j, T_{\text{sol}}))$$

$$= -p(\text{sim}, T_{\text{sol}}) \times \log_2(p(\text{sim}, T_{\text{sol}})) - p(\text{não}, T_{\text{sol}}) \times \log_2(p(\text{não}, T_{\text{sol}}))$$

$$= -\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right) = 0.97095$$

$$\text{info}(T_{\text{nublado}}) = -\sum_{j=1}^n p(C_j, T_{\text{nublado}}) \times \log_2(p(C_j, T_{\text{nublado}}))$$

$$= -p(\text{sim}, T_{\text{nublado}}) \times \log_2(p(\text{sim}, T_{\text{nublado}})) - p(\text{não}, T_{\text{nublado}}) \times \log_2(p(\text{não}, T_{\text{nublado}}))$$

$$= -\frac{4}{4} \times \log_2\left(\frac{4}{4}\right) - \frac{0}{4} \times \log_2\left(\frac{0}{4}\right) = 0$$

$$\text{info}(T_{\text{chuva}}) = -\sum_{j=1}^n p(C_j, T_{\text{chuva}}) \times \log_2(p(C_j, T_{\text{chuva}}))$$

$$= -p(\text{sim}, T_{\text{chuva}}) \times \log_2(p(\text{sim}, T_{\text{chuva}})) - p(\text{não}, T_{\text{chuva}}) \times \log_2(p(\text{não}, T_{\text{chuva}}))$$

$$= -\frac{3}{5} \times \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times \log_2\left(\frac{2}{5}\right) = 0.97095$$

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\text{info}(\text{Aparência}, T) = \sum_{i=1}^3 \frac{|T_i|}{|T|} \times \text{info}(T_i)$$

$$= \frac{|T_{\text{sol}}|}{|T|} \times \text{info}(T_{\text{sol}}) + \frac{|T_{\text{nublado}}|}{|T|} \times \text{info}(T_{\text{nublado}}) + \frac{|T_{\text{chuva}}|}{|T|} \times \text{info}(T_{\text{chuva}})$$

$$= \frac{5}{14} \times \text{info}(T_{\text{sol}}) + \frac{4}{14} \times \text{info}(T_{\text{nublado}}) + \frac{5}{14} \times \text{info}(T_{\text{chuva}})$$

$$= \frac{5}{14} \times \left(-\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right)\right)$$

$$+ \frac{4}{14} \times \left(-\frac{4}{4} \times \log_2\left(\frac{4}{4}\right) - \frac{0}{4} \times \log_2\left(\frac{0}{4}\right)\right)$$

$$+ \frac{5}{14} \times \left(-\frac{3}{5} \times \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times \log_2\left(\frac{2}{5}\right)\right)$$

$$= 0.69354 \text{ bits}$$

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\begin{aligned}
 \text{info(Temperatura, } T) &= \sum_{i=1}^3 \frac{|T_i|}{|T|} \times \text{info}(T_i) \\
 &= \frac{|T_{quente}|}{|T|} \times \text{info}(T_{quente}) + \frac{|T_{agradável}|}{|T|} \times \text{info}(T_{agradável}) + \frac{|T_{fria}|}{|T|} \times \text{info}(T_{fria}) \\
 &= \frac{4}{14} \times \left(-\frac{2}{4} \times \log_2\left(\frac{2}{4}\right) - \frac{2}{4} \times \log_2\left(\frac{2}{4}\right) \right) \\
 &\quad + \frac{6}{14} \times \left(-\frac{4}{6} \times \log_2\left(\frac{4}{6}\right) - \frac{2}{6} \times \log_2\left(\frac{2}{6}\right) \right) \\
 &\quad + \frac{4}{14} \times \left(-\frac{3}{4} \times \log_2\left(\frac{3}{4}\right) - \frac{1}{4} \times \log_2\left(\frac{1}{4}\right) \right) \\
 &= 0.91106 \text{ bits}
 \end{aligned}$$

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\begin{aligned}
 \text{info(Umidade, } T) &= \sum_{i=1}^2 \frac{|T_i|}{|T|} \times \text{info}(T_i) \\
 &= \frac{|T_{alta}|}{|T|} \times \text{info}(T_{alta}) + \frac{|T_{normal}|}{|T|} \times \text{info}(T_{normal}) \\
 &= \frac{7}{14} \times \left(-\frac{3}{7} \times \log_2\left(\frac{3}{7}\right) - \frac{4}{7} \times \log_2\left(\frac{4}{7}\right) \right) \\
 &\quad + \frac{7}{14} \times \left(-\frac{6}{7} \times \log_2\left(\frac{6}{7}\right) - \frac{1}{7} \times \log_2\left(\frac{1}{7}\right) \right) \\
 &= 0.78845 \text{ bits}
 \end{aligned}$$

Aparência	sim	não	Total	Temperatura	sim	não	Total	Umidade	sim	não	Total	Ventando	sim	não	Total	Jogar	
sol	2	3	5	quente	2	2	4	alta	3	4	7	falso	6	2	8	sim	9
nublado	4	0	4	agradável	4	2	6	normal	6	1	7	verdadeiro	3	3	6	não	5
chuva	3	2	5	fria	3	1	4										
Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	9	5	14	Total	14

$$\begin{aligned}
 \text{info(Ventando, } T) &= \sum_{i=1}^2 \frac{|T_i|}{|T|} \times \text{info}(T_i) \\
 &= \frac{|T_{falso}|}{|T|} \times \text{info}(T_{falso}) + \frac{|T_{verdadeiro}|}{|T|} \times \text{info}(T_{verdadeiro}) \\
 &= \frac{8}{14} \times \left(-\frac{6}{8} \times \log_2\left(\frac{6}{8}\right) - \frac{2}{8} \times \log_2\left(\frac{2}{8}\right) \right) \\
 &\quad + \frac{6}{14} \times \left(-\frac{3}{6} \times \log_2\left(\frac{3}{6}\right) - \frac{3}{6} \times \log_2\left(\frac{3}{6}\right) \right) \\
 &= 0.89216 \text{ bits}
 \end{aligned}$$

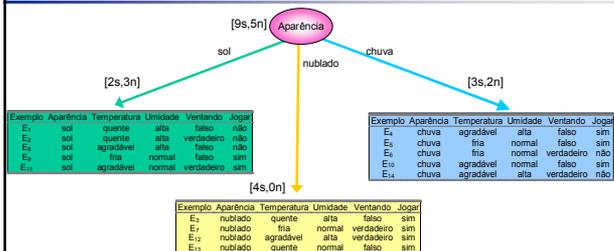
Escolha do Atributo para Particionar todo o Conjunto de Exemplos

$\text{info}(T) = 0.94029$ bits
 $\text{info}(Aparência, T) = 0.69354$ bits
 $\text{info}(Temperatura, T) = 0.91106$ bits
 $\text{info}(Umidade, T) = 0.78845$ bits
 $\text{info}(Ventando, T) = 0.89216$ bits

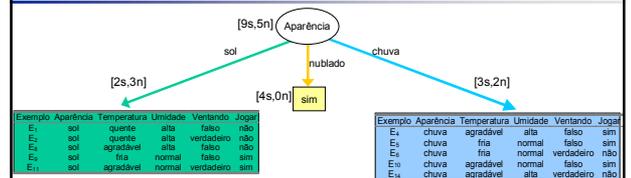
$\text{gain}(Aparência, T) = \text{info}(T) - \text{info}(Aparência, T) = 0.94029 - 0.69354 = 0.24675$ bits
 $\text{gain}(Temperatura, T) = \text{info}(T) - \text{info}(Temperatura, T) = 0.94029 - 0.91106 = 0.02922$ bits
 $\text{gain}(Umidade, T) = \text{info}(T) - \text{info}(Umidade, T) = 0.94029 - 0.78845 = 0.15184$ bits
 $\text{gain}(Ventando, T) = \text{info}(T) - \text{info}(Ventando, T) = 0.94029 - 0.89216 = 0.04813$ bits

$$\text{max-gain}(X, T) = \arg \max_{X \in \{X_1, X_2, \dots, X_m\}} \text{gain}(X, T) = \text{Aparência}$$

O Subconjunto "Aparência=nublado" possui Apenas Exemplos de uma Mesma Classe...



...o que Leva a um Nó Folha



Escolha do Atributo para Particionar "Aparência=sol"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.4$ bits
 $info(Umididade, T) = 0$ bits
 $info(Ventando, T) = 0.95098$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.4 = 0.57095$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0 = 0.97095$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Umididade$

Escolha do Atributo para Particionar "Aparência=sol"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.4$ bits
 $info(Umididade, T) = 0$ bits
 $info(Ventando, T) = 0.95098$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.4 = 0.57095$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0 = 0.97095$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Umididade$

Escolha do Atributo "Umididade" para Particionar "Aparência=sol"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.95098$ bits
 $info(Umididade, T) = 0.95098$ bits
 $info(Ventando, T) = 0$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0 = 0.97095$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Ventando$

Escolha do Atributo para Particionar "Aparência=chuva"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.95098$ bits
 $info(Umididade, T) = 0.95098$ bits
 $info(Ventando, T) = 0$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0 = 0.97095$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Ventando$

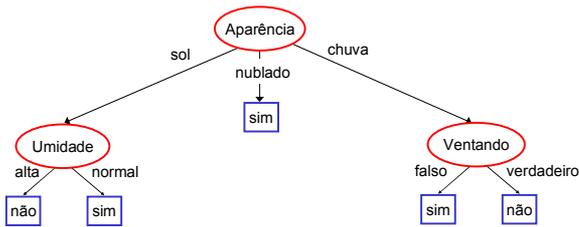
Escolha do Atributo para Particionar "Aparência=chuva"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.95098$ bits
 $info(Umididade, T) = 0.95098$ bits
 $info(Ventando, T) = 0$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0 = 0.97095$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Ventando$

Escolha do Atributo "Ventando" para Particionar "Aparência=chuva"

$inf(T) = 0.97095$ bits
 $info(Temperatura, T) = 0.95098$ bits
 $info(Umididade, T) = 0.95098$ bits
 $info(Ventando, T) = 0$ bits
 $gain(Temperatura, T) = info(T) - info(Temperatura, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Umididade, T) = info(T) - info(Umididade, T) = 0.97095 - 0.95098 = 0.01997$ bits
 $gain(Ventando, T) = info(T) - info(Ventando, T) = 0.97095 - 0 = 0.97095$ bits
 $max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Ventando$

Árvore de Decisão Induzida



62

Exercício

- Calcule o ganho para o atributo "Dia", ou seja, $\text{gain}(\text{Dia}, T)$, sabendo que $\text{info}(T) = 0.94$
 - $\text{gain}(\text{Dia}, T) = \text{info}(T) - \text{info}(\text{Dia}, T)$

Dia	Aparência	Temperatura	Umidade	Ventando	Jogar
d1	sol	quente	alta	falso	não
d2	sol	quente	alta	verdadeiro	não
d3	nublado	quente	alta	falso	sim
d4	chuva	agradável	alta	falso	sim
d5	chuva	fria	normal	falso	sim
d6	chuva	fria	normal	verdadeiro	não
d7	nublado	fria	normal	verdadeiro	sim
d8	sol	agradável	alta	falso	não
d9	sol	fria	normal	falso	sim
d10	chuva	agradável	normal	falso	sim
d11	sol	agradável	normal	verdadeiro	sim
d12	nublado	agradável	alta	verdadeiro	sim
d13	nublado	quente	normal	falso	sim
d14	chuva	agradável	alta	verdadeiro	não

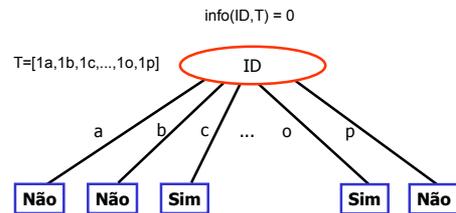
63

Razão de Ganho

- Vimos que o ganho máximo é interessante para particionar os exemplos, fornecendo bons resultados
- Entretanto, ele tem uma tendência (*bias*) em favor de testes com muitos valores
- Por exemplo, considere um conjunto de exemplos de diagnóstico médico no qual um dos atributos contém o código de identificação do paciente (ID)
- Uma vez que cada código ID é único, particionando o conjunto de treinamento nos valores deste atributo levará a um grande número de subconjuntos, cada um contendo somente um caso
- Como todos os subconjuntos (de 1 elemento) necessariamente contêm exemplos de uma mesma classe, $\text{info}(\text{ID}, T) = 0$, assim o ganho de informação deste atributo será máximo

64

Razão de Ganho



65

Razão de Ganho

- Para solucionar esta situação, em analogia à definição de $\text{info}(T)$, vamos definir a informação potencial gerada pela partição de T em r subconjuntos

$$\text{split-info}(X, T) = - \sum_{i=1}^r \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right)$$

- A razão de ganho é definida como:

$$\text{gain-ratio}(X, T) = \frac{\text{gain}(X, T)}{\text{split-info}(X, T)}$$

- A razão de ganho expressa a proporção de informação gerada pela partição que é útil, ou seja, que aparenta ser útil para a classificação

66

Razão de Ganho

- Usando o exemplo anterior para o atributo Aparência que produz três subconjuntos com 5, 4 e 5 exemplos, respectivamente

$$\text{split-info}(\text{Aparência}, T) = - \frac{5}{14} \times \log_2 \left(\frac{5}{14} \right) - \frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) - \frac{5}{14} \times \log_2 \left(\frac{5}{14} \right) = 1.57741 \text{ bits}$$

- Para este teste, cujo ganho é $\text{gain}(\text{Aparência}, T) = 0.24675$ (mesmo valor anterior), a razão de ganho é

- $\text{gain-ratio}(\text{Aparência}, T) = 0.24675 / 1.57741 = 0.156428$

67

Atributos Numéricos

- Se um atributo X assume valores reais (numéricos), é gerado um teste binário cujos resultados são $X \leq Z$ e $X > Z$
- O limite Z pode ser encontrado da seguinte forma
 - Os exemplos de T são inicialmente ordenados considerando os valores do atributo X sendo considerado
 - Há apenas um conjunto finito de valores, que podemos denotar (em ordem) por $\{V_1, V_2, \dots, V_J\}$
 - Qualquer limite caindo entre V_i e V_{i+1} tem o mesmo efeito que particionar os exemplos cujos valores do atributo X encontra-se em $\{V_1, V_2, \dots, V_i\}$ e em $\{V_{i+1}, V_{i+2}, \dots, V_J\}$
 - Assim, existem apenas $L-1$ divisões possíveis para o atributo X , cada uma devendo ser examinada
 - Isso pode ser obtido (uma vez ordenados os valores) em uma única passagem, atualizando as distribuições de classes para a esquerda e para a direita do limite Z durante o processo
 - Alguns indutores podem escolher o valor de limite como sendo o ponto médio de cada intervalo $Z = (V_i + V_{i+1})/2$
 - C4.5, entretanto, escolhe o maior valor de Z entre todo o conjunto de treinamento que não excede o ponto médio acima, assegurando que todos os valores que aparecem na árvore de fato ocorrem nos dados

Exemplo

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	nublado	83	86	falso	sim
E ₄	chuva	70	96	falso	sim
E ₅	chuva	68	80	falso	sim
E ₆	chuva	65	70	verdadeiro	não
E ₇	nublado	64	65	verdadeiro	sim
E ₈	sol	72	95	falso	não
E ₉	sol	69	70	falso	sim
E ₁₀	chuva	75	80	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim
E ₁₂	nublado	72	90	verdadeiro	sim
E ₁₃	nublado	81	75	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

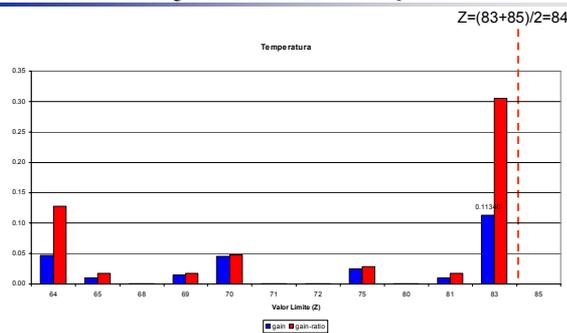
Escolha do Atributo para Particionar todo o Conjunto de Exemplos

Aparência	sim	não	Total	Ventando	sim	não	Total	Jogar
sol	2	3	5	falso	6	2	8	sim
nublado	4	0	4	verdadeiro	3	3	6	não
chuva	3	2	5					
Total	9	5	14	Total	9	5	14	Total

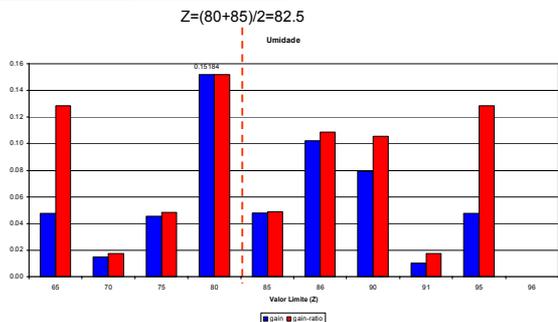
Temperatura	64	65	68	69	70	71	72	75	80	81	83	85
Jogar	sim	não	sim	sim	sim	não	não	sim	não	sim	sim	não

Umidade	65	70	75	80	85	86	90	91	95	96
Jogar	sim	não	sim	sim	não	sim	não	não	não	sim

Escolha do Atributo para Particionar todo o Conjunto de Exemplos



Escolha do Atributo para Particionar todo o Conjunto de Exemplos



Escolha do Atributo para Particionar todo o Conjunto de Exemplos

$info(T) = 0.94029$ bits
 $info(Aparência, T) = 0.69354$ bits
 $info(Temperatura_{Z=84}, T) = 0.93980$ bits
 $info(Umidade_{Z=82.5}, T) = 0.92997$ bits
 $info(Ventando, T) = 0.89216$ bits

 $gain(Aparência, T) = 0.24675$ bits
 $gain(Temperatura_{Z=84}, T) = 0.11340$ bits
 $gain(Umidade_{Z=82.5}, T) = 0.15184$ bits
 $gain(Ventando, T) = 0.04813$ bits

 $max-gain(X, T) = \arg \max_{X \in \{X_1, X_2, \dots, X_n\}} gain(X, T) = Aparência$

O Subconjunto "Aparência=nublado" possui Apenas Exemplos de uma Mesma Classe, Levando a um Nó Folha

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	sol	72	95	falso	não
E ₄	sol	69	70	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₅	chuva	70	90	falso	sim
E ₆	chuva	68	80	falso	sim
E ₇	chuva	65	70	verdadeiro	não
E ₁₀	chuva	75	80	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

Escolha do Atributo para Particionar "Aparência=sol"

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	sol	72	95	falso	não
E ₄	sol	69	70	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim

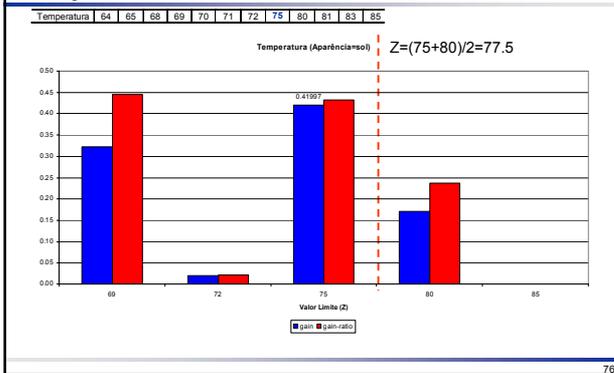
Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₅	chuva	70	90	falso	sim
E ₆	chuva	68	80	falso	sim
E ₇	chuva	65	70	verdadeiro	não
E ₁₀	chuva	75	80	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

Ventando	sim	não	Total	Jogar
falso	1	2	3	sim 2
verdadeiro	1	1	2	não 3
Total	2	3	5	Total 5

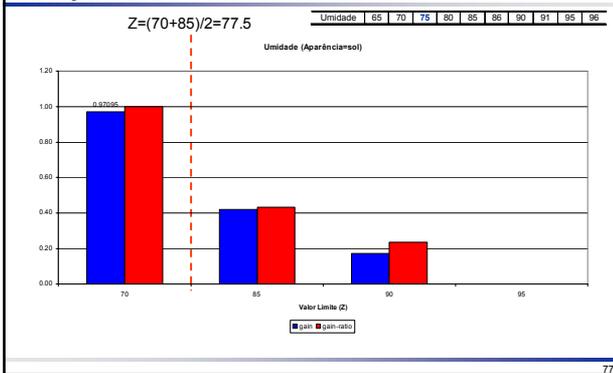
Temperatura	69	72	75	80	85
Jogar	sim	não	sim	não	não

Umidade	70	85	90	95
Jogar	sim	não	não	não

Escolha do Atributo para Particionar "Aparência=sol"



Escolha do Atributo para Particionar "Aparência=sol"



Escolha do Atributo para Particionar "Aparência=sol"

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	sol	72	95	falso	não
E ₄	sol	69	70	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₅	chuva	70	90	falso	sim
E ₆	chuva	68	80	falso	sim
E ₇	chuva	65	70	verdadeiro	não
E ₁₀	chuva	75	80	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

Ventando	sim	não	Total	Jogar
falso	1	2	3	sim 2
verdadeiro	1	1	2	não 3
Total	2	3	5	Total 5

Temperatura	69	72	75	80	85
Jogar	sim	não	sim	não	não

Umidade	70	85	90	95
Jogar	sim	não	não	não

$info(T) = 0.97095 \text{ bits}$
 $info(Temperatura_{Z=77.5}, T) = 0.55098 \text{ bits}$
 $info(Umidade_{Z=77.5}, T) = 0 \text{ bits}$
 $info(Ventando, T) = 0.95098 \text{ bits}$

$gain(Temperatura_{Z=77.5}, T) = 0.97095 - 0.55098 = 0.41997$
 $gain(Umidade_{Z=77.5}, T) = 0.97095 - 0 = 0.97095 \text{ bits}$
 $gain(Ventando, T) = 0.97095 - 0.95098 = 0.01997 \text{ bits}$
 $max_gain(X, T) = \arg \max_{X \in \{T, A_1, \dots, A_n\}} gain(X, T) = Umidade_{Z=77.5}$

Escolha do Atributo "Umidade" para Particionar "Aparência=sol"

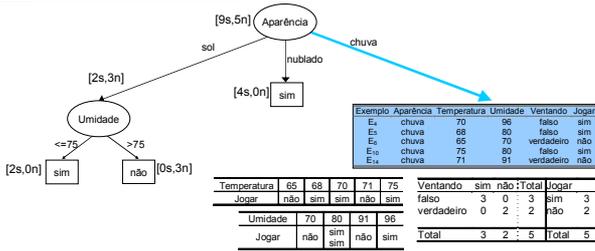
Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₉	sol	69	70	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	sol	72	95	falso	não

Ventando	sim	não	Total	Jogar
falso	1	1	2	sim 2
verdadeiro	1	0	1	não 1
Total	2	1	3	Total 3

Umidade	70	85	90	95
Jogar	sim	não	não	não

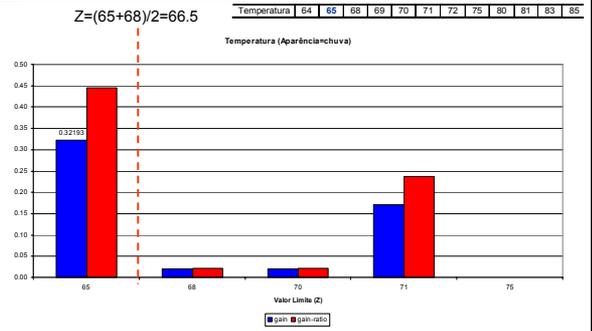
Escolha do Atributo para Particionar "Aparência=chuva"



80

Escolha do Atributo para Particionar "Aparência=chuva"

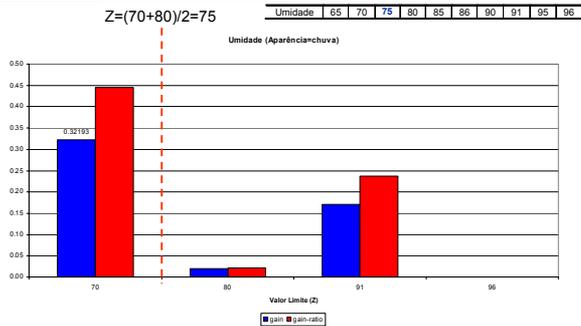
$$Z = (65+68)/2 = 66.5$$



81

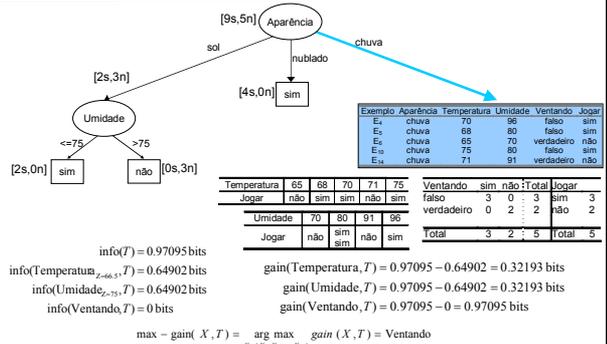
Escolha do Atributo para Particionar "Aparência=chuva"

$$Z = (70+80)/2 = 75$$



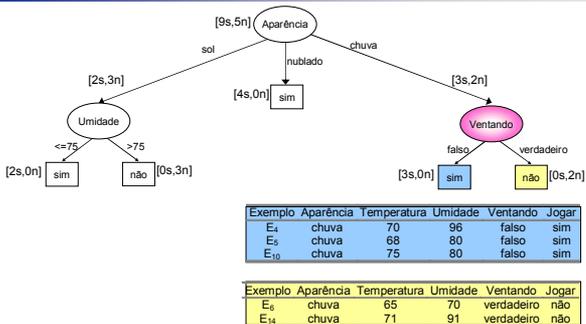
82

Escolha do Atributo para Particionar "Aparência=chuva"



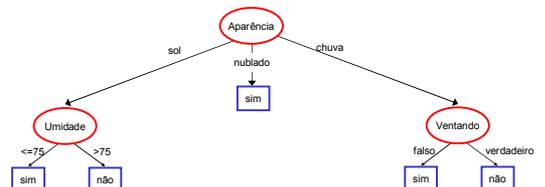
83

Escolha do Atributo "Ventando" para Particionar "Aparência=chuva"



84

Árvore de Decisão Induzida



85

Atributos com Valores Desconhecidos (Missing Values)

- ❑ O algoritmo básico para construção da DT assume que o valor de um teste para cada exemplo de treinamento possa ser determinado
- ❑ Além disso, o processo de classificação de novos exemplos requer uma escolha em cada ramo da árvore, escolha esta baseada em um atributo, cujo valor deve ser conhecido
- ❑ Entretanto, em dados do mundo real é freqüente o fato que um atributo apresente valores desconhecidos
 - O valor não é relevante para aquele exemplo particular
 - O valor não foi armazenado quando os exemplos foram coletados
 - O valor não pôde ser decifrado (se escrito à mão) pela pessoa que digitou os dados

86

Atributos com Valores Desconhecidos

- ❑ Por exemplo, Quinlan (1993) reporta que em um conjunto de 3000 dados médicos sobre tireóide, muitos exemplos não possuem o sexo do paciente, mesmo sabendo que esta informação seja usualmente relevante para a interpretação; mais de 30% dos exemplos apresentam valores desconhecidos
- ❑ Assim, a falta de completude é típica em dados do mundo real
- ❑ Diante disso, há algumas escolhas possíveis
 - Descartar uma parte (significante) dos exemplos de treinamento e assumir alguns dos novos exemplos (teste) como sendo inclassificáveis
 - Pré-processar os dados, substituindo os valores desconhecidos (o que geralmente altera o processo de aprendizado)
 - Alterar os algoritmos apropriadamente para tratar atributos contendo valores desconhecidos

87

Atributos com Valores Desconhecidos

- ❑ A alteração dos algoritmos para tratar atributos contendo valores desconhecidos requer a seguinte análise:
 - A escolha de um teste para particionar o conjunto de treinamento: se dois testes utilizam atributos com diferentes números de valores desconhecidos, qual o mais desejável?
 - Uma vez que um teste tenha sido escolhido, exemplos de treinamento com valores desconhecidos de um atributo não podem ser associados a um particular ramo (*outcome*) do teste e, portanto, não pode ser atribuído a um subconjunto particular T_j . Como esses exemplos devem ser tratados no particionamento?
 - Quando a árvore é utilizada para classificar um novo exemplo, como o classificador deve proceder se o exemplo tem um valor desconhecido para o atributo testado no nó de decisão atual?
- ❑ Veremos nos próximos slides a estratégia adotada pelo indutor C4.5

88

Escolha de um Teste

- ❑ Como mencionado, o ganho de informação de um teste mede a informação necessária para identificar uma classe que pode ser esperada por meio do particionamento do conjunto de exemplos, calculado como a subtração da informação esperada requerida para identificar a classe de um exemplo após o particionamento da mesma informação antes do particionamento
- ❑ É evidente que um teste não fornece informação alguma sobre a pertinência a uma classe de um exemplo cujo valor do atributo de teste é desconhecido

89

Escolha de um Teste

- ❑ Assumindo que uma fração F de exemplos tenha seu valor conhecido para o atributo X , a definição de ganho pode ser alterada para
 - $gain(X, T) = \text{probabilidade de } X \text{ ser conhecido} * (\text{info}(T) - \text{info}(X, T))$
+ probabilidade de X ser desconhecido * 0
 - $gain(X, T) = F * (\text{info}(T) - \text{info}(X, T))$
- ❑ De forma similar, a definição de $split\text{-info}(X, T)$ pode ser alterada considerando os exemplos com valores desconhecidos como um grupo adicional. Se o teste tem r valores, seu $split\text{-info}$ é calculado como se o teste dividisse os exemplos em $r+1$ subconjuntos

90

Exercício

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	nublado	83	86	falso	sim
E ₄	chuva	70	96	falso	sim
E ₅	chuva	68	80	falso	sim
E ₆	chuva	65	70	verdadeiro	não
E ₇	nublado	64	65	verdadeiro	sim
E ₈	sol	72	95	falso	não
E ₉	sol	69	70	falso	sim
E ₁₀	chuva	75	80	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim
E ₁₂	nublado	72	90	verdadeiro	sim
E ₁₃	nublado	81	75	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

91

Exercício

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar
E ₁	sol	85	85	falso	não
E ₂	sol	80	90	verdadeiro	não
E ₃	nublado	83	86	falso	sim
E ₄	chuva	70	96	falso	sim
E ₅	chuva	68	80	falso	sim
E ₆	chuva	65	70	verdadeiro	não
E ₇	nublado	64	65	verdadeiro	sim
E ₈	sol	72	95	falso	não
E ₉	sol	69	70	falso	sim
E ₁₀	chuva	75	80	falso	sim
E ₁₁	sol	75	70	verdadeiro	sim
E ₁₂	?	72	90	verdadeiro	sim
E ₁₃	nublado	81	75	falso	sim
E ₁₄	chuva	71	91	verdadeiro	não

Calcular $\text{info}(T)$, $\text{info}(\text{Aparência}, T)$, $\text{gain}(\text{Aparência}, T)$, $\text{split-info}(\text{Aparência}, T)$, $\text{gain-ratio}(\text{Aparência}, T)$

92

Solução

Aparência	sim	não	Total
sol	2	3	5
nublado	3	0	3
chuva	3	2	5
Total	8	5	13

$$\text{info}(T) = -\frac{8}{13} \times \log_2\left(\frac{8}{13}\right) - \frac{5}{13} \times \log_2\left(\frac{5}{13}\right) = 0.9612 \text{ bits}$$

$$\begin{aligned} \text{info}(\text{Aparência}, T) &= \frac{5}{13} \times \left(-\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right) \right) \\ &\quad + \frac{3}{13} \times \left(-\frac{3}{3} \times \log_2\left(\frac{3}{3}\right) - \frac{0}{3} \times \log_2\left(\frac{0}{3}\right) \right) \\ &\quad + \frac{5}{13} \times \left(-\frac{3}{5} \times \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times \log_2\left(\frac{2}{5}\right) \right) \\ &= 0.7469 \end{aligned}$$

$$\begin{aligned} \text{gain}(\text{Aparência}, T) &= \frac{13}{14} \times (0.9612 - 0.7469) \\ &= 0.1990 \text{ bits} \end{aligned}$$

93

Solução

Aparência	sim	não	Total
sol	2	3	5
nublado	3	0	3
chuva	3	2	5
Total	8	5	13

$$\begin{aligned} \text{split-info}(\text{Aparência}, T) &= -\frac{5}{14} \times \log_2\left(\frac{5}{14}\right) \quad (\text{para sol}) \\ &\quad - \frac{3}{14} \times \log_2\left(\frac{3}{14}\right) \quad (\text{para nublado}) \\ &\quad - \frac{5}{14} \times \log_2\left(\frac{4}{14}\right) \quad (\text{para chuva}) \\ &\quad - \frac{1}{14} \times \log_2\left(\frac{1}{14}\right) \quad (\text{para ?}) \\ &= 1.8092 \end{aligned}$$

$$\text{gain-ratio}(\text{Aparência}, T) = \frac{0.1990}{1.8092} = 0.1100$$

94

Particionando o Conjunto de Treinamento

- Um teste pode ser selecionado dentre os possíveis testes como antes, utilizando as definições modificadas de *gain* e *split-info*
- Se o atributo selecionado X possui valores desconhecidos, o conceito de particionamento do conjunto T é generalizado da seguinte forma:
 - Assumindo que X assume r valores, ou seja $X = O_1, X = O_2, \dots, X = O_r$, cada teste particiona o conjunto T nos subconjuntos T_1, T_2, \dots, T_r , respectivamente
 - Quando um exemplo de T com valor conhecido é atribuído ao subconjunto T_i , isto indica que a probabilidade daquele exemplo pertencer ao subconjunto T_i é 1 e em todos os demais subconjuntos é 0

95

Particionando o Conjunto de Treinamento

- Quando um exemplo possui valor desconhecido, apenas um grau de pertinência probabilístico pode ser feito
- Assim a cada exemplo em cada subconjunto T_i é associado um **peso** representando a probabilidade do exemplo pertencer a cada subconjunto
 - Se o exemplo tem seu valor conhecido para o teste, o peso é 1
 - Se o exemplo tem seu valor desconhecido para o teste, o peso é a probabilidade do teste $X=O_i$ naquele ponto; cada subconjunto T_i é agora uma coleção de exemplos fracionários de forma que $|T_i|$ deve ser interpretado como a soma dos pesos fracionários dos exemplos no subconjunto

96

Particionando o Conjunto de Treinamento

- Os exemplos em T podem ter pesos não unitários, uma vez que T pode ser um subconjunto de uma partição anterior
- Em geral, um exemplo de T com peso w cujo valor de teste é desconhecido é atribuído a cada subconjunto T_i com peso
 - $w * \text{probabilidade de } X=O_i$
- A probabilidade é estimada como a soma dos pesos dos exemplos em T que têm seu valor (conhecido) igual a O_i dividido pela soma dos pesos dos exemplos em T que possuem valores conhecidos para o atributo X

97

Exemplo

- Quando os 14 exemplos são particionados pelo atributo **Aparência**, os 13 exemplos para os quais o valor é conhecido não apresentam problemas
- O exemplo remanescente é atribuído para todas as partições, correspondendo aos valores **sol**, **nublado** e **chuva**, com pesos 5/13, 3/13 e 5/13, respectivamente

98

Exemplo

- Vamos analisar a primeira partição, correspondendo a Aparência=sol

Exemplo	Aparência	Temperatura	Umidade	Ventando	Jogar	Peso
E ₁	sol	85	85	falso	não	1
E ₂	sol	80	90	verdadeiro	não	1
E ₃	sol	72	95	falso	não	1
E ₉	sol	69	70	falso	sim	1
E ₁₁	sol	75	70	verdadeiro	sim	1
E ₁₂	?	72	90	verdadeiro	sim	5/13

- Se este subconjunto for particionado novamente pelo mesmo teste anterior, ou seja, utilizando o atributo **Umidade**, teremos as seguintes distribuições de classes
 - Umidade \leq 75 [2s, 0n]
 - Umidade $>$ 75 [5/13s, 3n]

99

Exemplo

- Distribuições de classes
 - Umidade \leq 75 [2s, 0n]
 - Umidade $>$ 75 [5/13s, 3n]
- A primeira partição contém exemplos de uma única classe (**sim**)
- A segunda ainda contém exemplos de ambas as classes mas o algoritmo não encontra nenhum teste que melhore sensivelmente esta situação
- De maneira similar, o subconjunto correspondendo a Aparência=chuva e cujo teste esteja baseado no atributo **Ventando** (como anteriormente) não pode ser particionado em subconjuntos de uma única classe

100

Exemplo

- A DT assume a forma:

```
aparencia = sol
...umidade <= 75: sim (2.0)
: umidade > 75: não (3.4/0.4)
aparencia = nublado: sim (3.2)
aparencia = chuva
...ventando = verdadeiro: não (2.4/0.4)
: ventando = falso: sim (3.0)
```

- Os número nas folhas da forma (N) ou (N/E) significam
 - N é a soma de exemplos fracionários que atingiram a folha
 - E é o número de exemplos que pertencem a classes diferentes daquela predita pela folha (em árvores não podadas)

101

Classificando um Novo Exemplo

- Uma abordagem similar é utilizada quando a DT é usada para classificar um novo exemplo
- Se um nó de decisão é encontrado para o qual o valor do atributo é desconhecido (ou seja, o valor do teste não pode ser determinado), o algoritmo explora todos os valores possíveis de teste, combinando o resultado das classificações aritmeticamente
- Uma vez que agora podem haver múltiplos caminhos da raiz da árvore ou sub-árvore até as folhas, a "classificação" é uma distribuição de classes ao invés de uma única classe
- Quando a distribuição total de classes para o novo exemplo é estabelecida, a classe com a maior probabilidade é rotulada como sendo "a" classe predita

102

Exemplo

Aparência	Temperatura	Umidade	Ventando
sol	75	?	falso

```
aparencia = sol
...umidade <= 75: sim (2.0)
: umidade > 75: não (3.4/0.4)
aparencia = nublado: sim (3.2)
aparencia = chuva
...ventando = verdadeiro: não (2.4/0.4)
: ventando = falso: sim (3.0)
```

- O valor de **Aparência** assegura que o exemplo mova-se para a primeira sub-árvore mas não é possível determinar se **Umidade** \leq 75
- Entretanto, podemos notar que:
 - Se Umidade \leq 75 o exemplo poderia ser classificado como **sim**
 - Se Umidade $>$ 75, o exemplo poderia ser classificado como **não** com probabilidade 3/3.4 (88%) e **sim** com probabilidade 0.4/3.4 (12%)
- Quando a DT foi construída, as partições para estes testes tinham 2.0 e 3.4 exemplos, respectivamente
- As conclusões condicionais são combinadas com os mesmos pesos relativos 2.0/5.4 e 3.4/5.4 de forma que a distribuição final de classes para o exemplo é
 - sim: $2.0/5.4 * 100\% + 3.4/5.4 * 12\% = 44\%$
 - não: $3.4/5.4 * 88\% = 56\%$

103

Poda

- ❑ Há duas formas de produzir árvores mais simples
 - **pré-poda:** decide-se não mais particionar o conjunto de treinamento, utilizando algum critério
 - **pós-poda:** induz-se a árvore completa e então remove-se alguns dos ramos
- ❑ A poda invariavelmente causará a classificação incorreta de exemplos de treinamento
- ❑ Conseqüentemente, as folhas não necessariamente conterão exemplos de uma única classe

104

Pré-Poda

- ❑ Evita gastar tempo construindo estruturas (sub-árvores) que não serão usadas na árvore final simplificada
- ❑ O método usual consiste em analisar a melhor forma de particionar um subconjunto, mensurando-a sob o ponto de vista de significância estatística, ganho de informação, redução de erro ou outra métrica qualquer
- ❑ Se a medida encontrada encontrar-se abaixo de um valor limite (*threshold*) o particionamento é interrompido e a árvore para aquele subconjunto é apenas a folha mais apropriada
- ❑ Entretanto, a definição do valor limite não é simples de ser definido
 - Um valor muito grande pode terminar o particionamento antes que os benefícios de divisões subseqüentes tornem-se evidentes
 - Um valor muito pequeno resulta em pouca simplificação

105

Pós-Poda

- ❑ O processo de indução (*dividir-e-conquistar*) da árvore continua de forma livre e então a árvore super-ajustada (*overfitted tree*) produzida é então podada
- ❑ O custo computacional adicional investido na construção de partes da árvore que serão posteriormente descartadas pode ser substancial
- ❑ Entretanto, esse custo é compensador devido a uma maior exploração das possíveis partições
- ❑ Crescer e podar árvores é mais lento, mas mais confiável

106

Pós-Poda

- ❑ Existem várias forma de avaliar a taxa de erro de árvores podadas, dentre elas
 - avaliar o desempenho em um subconjunto separado do conjunto de treinamento (o que implica que uma parte dos exemplos devem ser reservada para a poda e, portanto, a árvore tem que ser construída a partir de um conjunto de exemplos menor)
 - avaliar o desempenho no conjunto de treinamento, mas ajustando o valor estimado do erro, já que ele tem a tendência de ser menor no conjunto de treinamento

107

Pós-Poda (C4.5)

- ❑ Quando N exemplos de treinamento são cobertos por uma folha, E dos quais incorretamente, a taxa de erro de substituição para esta folha é E/N
- ❑ Entretanto, isso pode ser visto como a observação de E eventos em N tentativas
- ❑ Se esse conjunto de N exemplos de treinamento forem vistos como uma amostra (o que de fato não é), podemos analisar o que este resultado indica sobre a probabilidade de um evento (erro) na população inteira de exemplos cobertos por aquela folha
- ❑ A probabilidade não pode ser determinada exatamente, mas tem uma distribuição de probabilidade (posterior) que é usualmente resumida por um par de limites de confiança
- ❑ Para um dado nível de confiança CF , o limite superior desta probabilidade pode ser encontrado a partir dos limites de confiança de uma distribuição binomial denotado por $U_{CF}(E,N)$
- ❑ Uma folha com N exemplos de treinamento com uma taxa de erro predita de $U_{CF}(E,N)$ totalizará $N \cdot U_{CF}(E,N)$ erros

108

Análise de Complexidade

- ❑ Vamos assumir que a profundidade da árvore para n exemplos é $O(\log n)$ (assumindo árvore balanceada)
- ❑ Vamos considerar o esforço para um atributo para todos os nós da árvore; nem todos os exemplos precisam ser considerados em cada nó mas certamente o conjunto completo de n exemplos deve ser considerado em cada nível da árvore
- ❑ Como há $\log n$ níveis na árvore, o esforço para um único atributo é $O(n \log n)$
- ❑ Assumindo que em cada nó todos os atributos são considerados, o esforço para construir a árvore torna-se $O(mn \log n)$
 - Se os atributos são numéricos, eles devem ser ordenados, mas apenas uma ordenação inicial é necessária, o que toma $O(n \log n)$ para cada um dos m atributos: assim a complexidade acima permanece a mesma
 - Se os atributos são nominais, nem todos os atributos precisam ser considerados em cada nó uma vez que atributos utilizados anteriormente não podem ser reutilizados; entretanto, se os atributos são numéricos eles podem ser reutilizados e, portanto, eles devem ser considerados em cada nível da árvore

113

Análise de Complexidade

- Na poda (*subtree replacement*), inicialmente uma estimativa de erro deve ser efetuada em cada nó
 - Assumindo que contadores sejam apropriadamente mantidos, isto é realizado em tempo linear ao número de nós na árvore
- Após isso, cada nó deve ser considerado para substituição
 - A árvore possui no máximo n folhas, uma para cada exemplo
 - Se a árvore for binária (cada atributo sendo numérico ou nominal com dois valores apenas) isso resulta em $2n-1$ nós (árvores com multi-ramos apenas diminuem o número de nós internos)
- Assim, a complexidade para a poda é $O(n)$

114

Interpretação Geométrica

- Consideramos exemplos como um vetor de m atributos
- Cada vetor corresponde a um ponto em um espaço m -dimensional
- A DT corresponde a uma divisão do espaço em regiões, cada região rotulada como uma classe

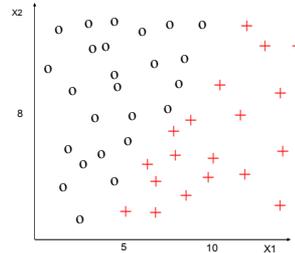
115

Interpretação Geométrica: Atributo-Valor

- Um teste para um atributo é da forma $X_i \text{ op Valor}$ onde X_i é um atributo, $op \in \{=, \neq, <, \leq, >, \geq\}$ e valor é uma constante válida para o atributo
- O espaço de descrição é particionado em regiões retangulares, nomeadas hiperplanos, que são ortogonais aos eixos
- As regiões produzidas por DT são todas hiperplanos
- Enquanto a árvore está sendo formada, mais regiões são adicionadas ao espaço

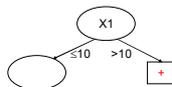
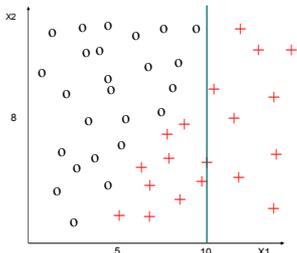
116

Interpretação Geométrica p/ DT



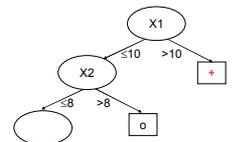
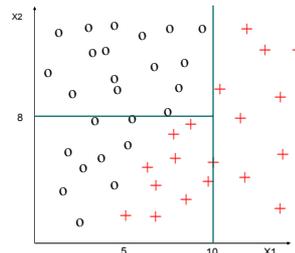
117

Interpretação Geométrica p/ DT



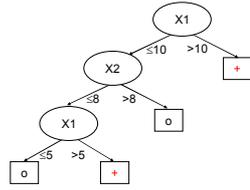
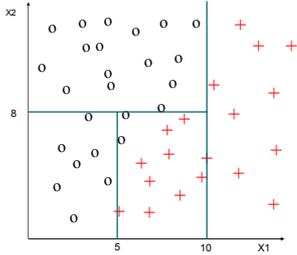
118

Interpretação Geométrica p/ DT



119

Interpretação Geométrica p/ DT



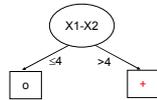
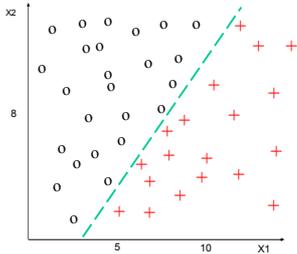
120

Combinação Linear de Atributos

- Produzem árvores de decisão oblíquas
- A representação para os testes são da forma
$$a_1 \times X_1 + a_2 \times X_2 + \dots + a_m \times X_m \text{ op Valor}$$
onde a_i é uma constante, X_i é um atributo real, $op \in \{<, \leq, >, \geq\}$ e $Valor$ uma constante
- O espaço de descrição é particionado hiperplanos que não são necessariamente ortogonais ao eixos

121

Árvore de Decisão Oblíqua



122

Resumo

- Árvores de decisão, em geral, possuem um tempo de aprendizado relativamente rápido
- Árvores de decisão permitem a classificação de conjuntos com milhões de exemplos e centenas de atributos a uma velocidade razoável
- É possível converter para regras de classificação, podendo ser interpretadas por seres humanos
- Precisão comparável a outros métodos

123