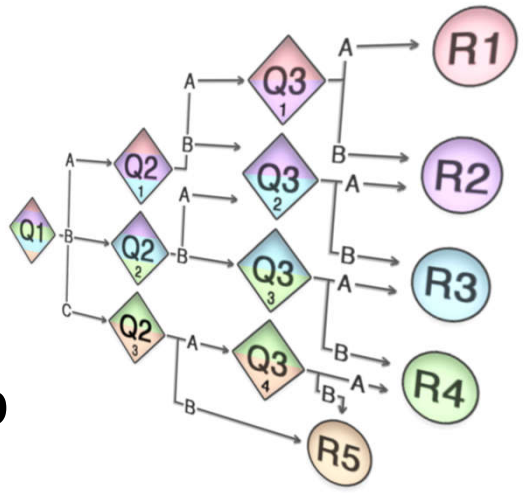
Aprendizado Simbólico:

ÁRVORES DE DECISÃO



Árvores de Decisão (ADs)

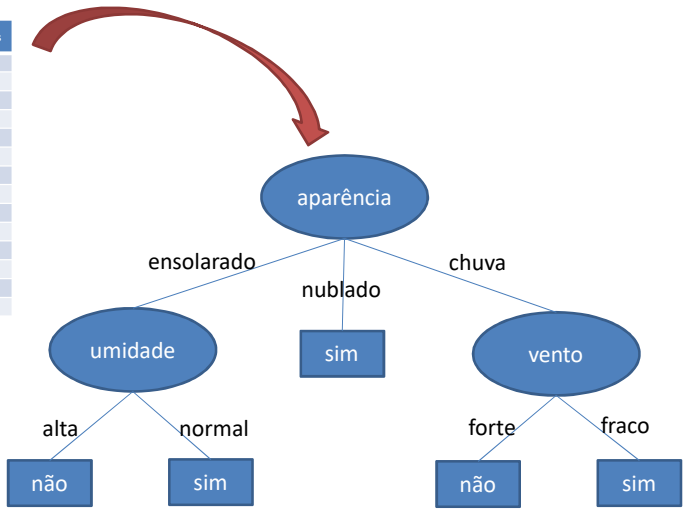
- Um dos métodos mais usados e práticos para inferência indutiva
  - Conhecimento estrutural do domínio
- Indução a partir de um conjunto de dados rotulados (classificados)
  - Aprendizado supervisionado
- Algumas áreas de aplicação: medicina (diagnóstico médico) e análise de risco de crédito para novos clientes de banco

Exemplo

Dia	Aparência	Temperatura	Umidade	Vento	Jogar tênis
D1	ensolarado	quente	alta	fraco	**não**
D2	ensolarado	quente	alta	forte	**não**
D3	nublado	quente	alta	fraco	**sim**
D4	chuva	moderada	alta	fraco	**sim**
D5	chuva	fria	normal	fraco	**sim**
D6	chuva	fria	normal	forte	**não**
D7	nublado	fria	normal	forte	**sim**
D8	ensolarado	moderada	alta	fraco	**não**
D9	ensolarado	fria	normal	fraco	**sim**
D10	chuva	moderada	normal	fraco	**sim**
D11	ensolarado	moderada	normal	forte	**sim**
D12	nublado	moderada	alta	forte	**sim**
D13	nublado	quente	normal	fraco	**sim**
D14	chuva	moderada	alta	forte	**não**

Exemplo

Dia	Aparência	Temperatura	Umidade	Vento	Jogar tênis
D1	ensolarado	quente	alta	fraco	**não**
D2	ensolarado	quente	alta	forte	**não**
D3	nublado	quente	alta	fraco	**sim**
D4	chuva	moderada	alta	fraco	**sim**
D5	chuva	fria	normal	fraco	**sim**
D6	chuva	fria	normal	forte	**não**
D7	nublado	fria	normal	forte	**sim**
D8	ensolarado	moderada	alta	fraco	**não**
D9	ensolarado	fria	normal	fraco	**sim**
D10	chuva	moderada	normal	fraco	**sim**
D11	ensolarado	moderada	normal	forte	**sim**
D12	nublado	moderada	alta	forte	**sim**
D13	nublado	quente	normal	fraco	**sim**
D14	chuva	moderada	alta	forte	**não**



Propriedades das ADs

- Instâncias (exemplos) são representados por pares atributo-valor.
- A hipótese/função objetivo (classe) tem, preferencialmente, valores discretos.
  - Atributos contínuos podem ser usados fazendo o nó dividir o domínio do atributo entre dois intervalos baseados em um limite (ex. tamanho < 3 e tamanho ≥3)
  - Árvores de classificação têm valores discretos nas folhas, árvores de regressão têm valores reais nas folhas.
- Algoritmos para encontrar árvores consistentes são eficientes e podem processar grandes quantidades de dados de treinamento

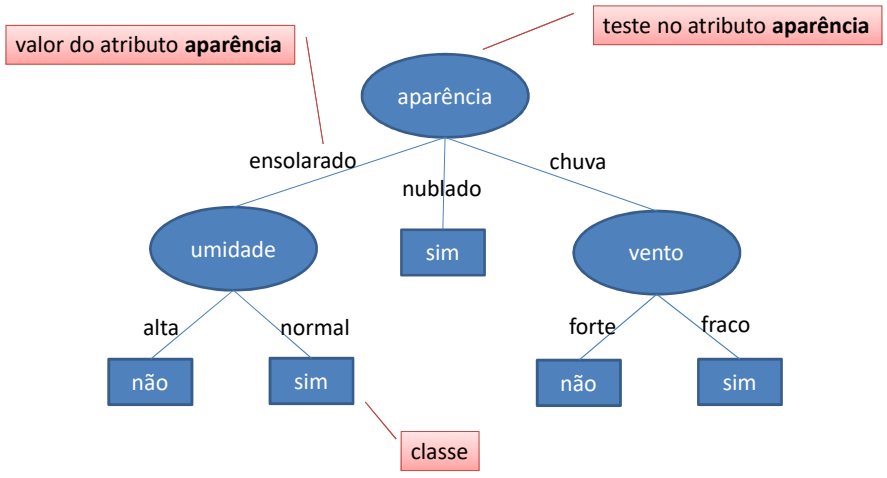
Propriedades das ADs

- Descrições disjuntivas podem ser necessárias.
- Os métodos de aprendizado de árvore de decisão lidam bem com ruído.
  - O conjunto de treinamento pode conter dados problemáticos: valores errados, incompletos ou inconsistentes.
- Existem métodos de árvore de decisão que permitem valores desconhecidos para algumas características.

Estrutura

- Uma árvore de decisão contém:
- Nós-folha que correspondem às classes.
- Nós de decisão que contêm testes sobre atributos.
  - Para cada resultado de um teste, existe uma aresta para uma sub-árvore; cada sub-árvore tem a mesma estrutura que a árvore.

Exemplo

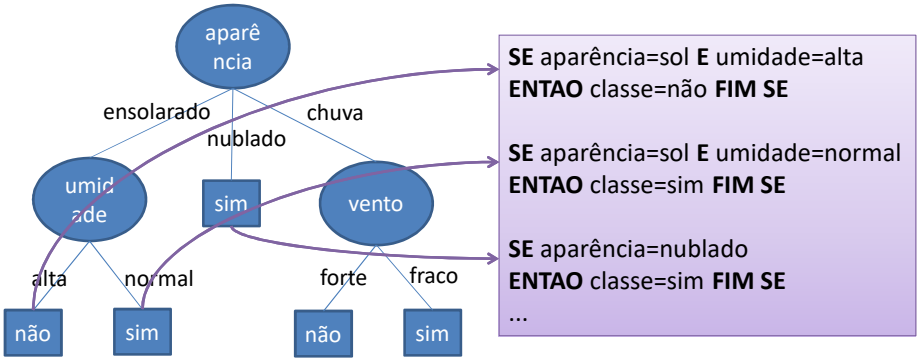


Exemplo

- Para classificar (dizer se vai chover), basta começar pela raiz, seguindo cada teste até que a folha seja alcançada.

Transformação em Regras

- Toda AD pode ser reescrita como um conjunto de regras, por exemplo, em forma normal disjuntiva (DNF).
  - Cada regra é obtida percorrendo a AD da raiz até as folhas.



Algoritmos de Aprendizado de AD

- Muitos algoritmos foram desenvolvidos para aprendizado de AD.
  - CART (1977)
  - ID3 (1979)
  - ASSISTANT (1984)
  - C4.5 (1993), aprimoramento do ID3
- A maioria utiliza abordagem top-down e busca gulosa no espaço de possíveis árvores de decisão.

Construindo uma AD

1. O conjunto de treinamento T contém um ou mais exemplos, todos da mesma classe Cj: AD para T é um nó-folha identificando a classe Cj.
2. T não contém exemplos: AD é um nó-folha, mas a classe associada à folha é determinada por outras informações sobre T.
3. T contém exemplos de diversas classes: refinar T em subconjuntos de exemplos que são (ou possam vir a ser) conjunto de exemplos de uma única classe; a divisão em subconjuntos é feita em função de valores de atributos.
4. Aplicar os passos 1, 2 e 3 recursivamente para cada subconjunto de exemplos de treinamento até que o i-ésimo ramo conduza a uma AD construída sobre o subconjunto Ti do conjunto de treinamento.
5. Depois de construída a AD, utilizar técnicas de poda.

Atributo de Particionamento

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

Atributo de Particionamento

- Existem várias possibilidades para a escolha do atributo:
  - **Aleatória:** seleciona um atributo aleatoriamente
  - **Menos valores:** escolhe o atributo com o menor número de valores possíveis
  - **Mais valores:** escolhe o atributo com o maior número de valores possíveis
  - **Ganho Máximo:** seleciona o atributo que possui o maior ganho de informação esperado, isto é, escolhe o atributo que resultará no menor tamanho para as sub-árvores "enraizadas" em seus filhos
  - **Índice Gini** (Breiman et al, 1984)
  - **Razão de ganho** (Quinlan, 1993)

Algoritmo ID3

- O algoritmo ID3 (*Inductive Decision Tree*) é um dos mais utilizados para a construção de ADs.
  - Um dos mais famosos, proposto por Quinlan.
- Características:
  - Top-down
  - Busca gulosa no espaço de possíveis AD
  - Medida para selecionar atributos: ganho de informação.
    - A medida de ganho de informação é usada para selecionar o atributo de teste, entre os atributos candidatos, em cada passo do crescimento da árvore.

Algoritmo ID3

- O ID3 começa a construção da árvore com a pergunta:
  - Qual atributo deveria ser testado como raiz da árvore?
- Para saber a resposta, usa a medida estatística ganho de informação, a qual mede quão bem um dado atributo separa o conjunto de treinamento de acordo com a classe.
- Árvore é construída recursivamente de cima para baixo, usando divisão e conquista.

Algoritmo ID3

1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja, que agrupa exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
4. Transportar os exemplos para cada filho tendo em conta o valor do filho;
5. Repetir o procedimento para cada filho não "puro". Um filho é puro quando cada atributo X tem o mesmo valor em todos os exemplos.

Pseudocódigo

```
(01) função DTree(exemplos, atributos): retorna uma árvore
(02) início
(03)   se todos exemplos pertencem a uma única classe então
(04)     retorna um nó folha com essa classe;
(05)   senão
(06)     se o conjunto de atributos estiver vazio então
(07)       retorna um nó folha com a classe mais comum entre os exemplos.
(08)     senão
(09)       escolha um atributo F e crie um nó R para ele
(10)       para cada possível valor v_i de F faça
(11)         seja exemplos_i o subconjunto de exemplos que tenha valor v_i para F
(12)         coloque uma aresta E a partir do nó R com o valor v_i.
(13)         se exemplos_i estiver vazio então
(14)           coloque uma folha ligado a aresta E com a classe mais comum entre
                                                                    os exemplos.
(15)         senão
(16)           chame DTree(exemplos_i, atributos – {F}) e ligue a árvore resultante
                                                        como uma sub-árvore sob E.
(17)     retorne a sub-árvore com raiz R.
(18) fim.
```

Ganho de Informação

- A medida de ganho de informação pode ser definida como a redução esperada na entropia causada pelo particionamento de exemplos de acordo com um determinado atributo.
  - Entropia: grau de desordem/surpresa de um conjunto de dados
    - Quanto menor a entropia, mais previsível e organizado é o conjunto de dados.
  - Original da Teoria da Informação, para calcular o **número de bits** necessários para a codificação de uma mensagem.
    - Quanto menor a entropia, menos bits são necessários para codificar a mensagem.

Entropia

- Entropia (desordem, impureza) de um conjunto de exemplos, S, relativa a classificação binária é:

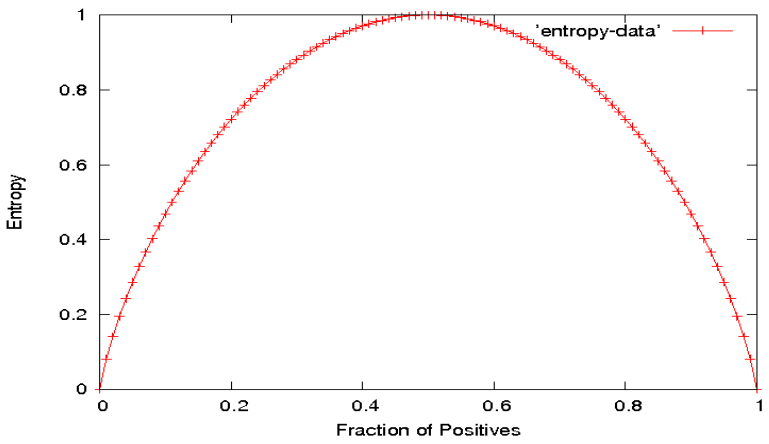
$$Entropy(S) = -p_{(+)} \log_2(p_{(+)}) - p_{(-)} \log_2(p_{(-)})$$

onde $p_{(+)}$ é a fração de exemplos positivos em S e $p_{(-)}$ é a fração de negativos.

- Genericamente, para qualquer número c de classes de um conjunto de dados, a entropia de S é dada pela fórmula:

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

em que p_i é a proporção de instâncias (exemplos) de S pertencendo a classe i e c é o número total de classes.



Entropia para Classificação Binária

- Se todos os exemplos forem da mesma classe, a entropia é zero (definimos $0.\log(0) = 0$)
- Se os exemplos estiverem igualmente misturados ($p_1 = p_0 = 0.5$), entropia é 1.
- Se a coleção contém número diferente de exemplos positivos e negativos, a entropia varia entre 0 e 1.

Exemplo

- Dada uma coleção S com 14 exemplos, sendo que o atributo de classe é constituído por 9 casos positivos e 5 casos negativos, $[9+, 5-]$, a entropia de S é:
- $Entropy(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$
- $Entropy(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$
- $Entropy(S) = 0.940$

Ganho de Informação

- A medida de ganho de informação pode ser definida como a redução esperada na entropia causada pelo particionamento de exemplos de acordo com um determinado atributo.
  - O **ganho de informação** deve ser calculado para cada atributo do conjunto de atributos da coleção S.
  - O atributo que resultar no **maior ganho de informação** é selecionado como atributo de teste.

Ganho de Informação

- O ganho de informação de um atributo A relativo à coleção S é definido como:

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

- em que $Valores(A)$ é o conjunto de todos os possíveis valores de A, S_v é o subconjunto de S para o qual o atributo A tem valor v, o primeiro termo é a entropia da coleção S e o segundo termo é a soma das entropias de cada valor presente no atributo A.

Ganho de Informação – Exemplo

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

Ganho de Informação – Exemplo

Dados: Valores(Vento) = fraco, forte Distribuição de S_{fraco} = [6+, 2-] Distribuição de S = [9+, 5-]
Distribuição de S_{forte} = [3+, 3-]

- $Gain(S, Vento) = Entropy(S) - \sum_{v \in \{Fraco, Forte\}} \frac{|S_v|}{|S|} Entropy(S_v)$
- $Gain(S, Vento) = Entropy(S) - \left(-\frac{6}{8} log_2\left(\frac{6}{8}\right) - \frac{2}{8} log_2\left(\frac{2}{8}\right)\right) \frac{|S_v|}{|S|} Entropy(S_v) - \left(-\frac{3}{6} log_2\left(\frac{3}{6}\right) - \frac{3}{6} log_2\left(\frac{3}{6}\right)\right)$
- $Gain(S, Vento) = 0.940 - \frac{8}{14} Entropy(S_{fraco}) - \frac{6}{14} Entropy(S_{forte})$
- $Gain(S, Vento) = 0.940 - \frac{8}{14} 0.811 - \frac{6}{14} 1.00$
- $Gain(S, Vento) = 0.048$

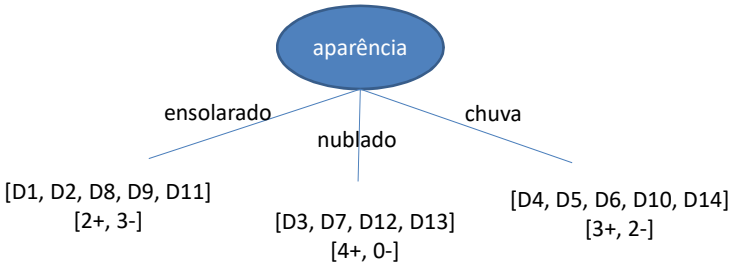
Construindo uma AD

Dia	Aparência	Temperatura	Umidade	Vento	Jogar tênis
D1	ensolarado	quente	alta	fraco	não
D2	ensolarado	quente	alta	forte	não
D3	nublado	quente	alta	fraco	sim
D4	chuva	moderada	alta	fraco	sim
D5	chuva	fria	normal	fraco	sim
D6	chuva	fria	normal	forte	não
D7	nublado	fria	normal	forte	sim
D8	ensolarado	moderada	alta	fraco	não
D9	ensolarado	fria	normal	fraco	sim
D10	chuva	moderada	normal	fraco	sim
D11	ensolarado	moderada	normal	forte	sim
D12	nublado	moderada	alta	forte	sim
D13	nublado	quente	normal	fraco	sim
D14	chuva	moderada	alta	forte	não

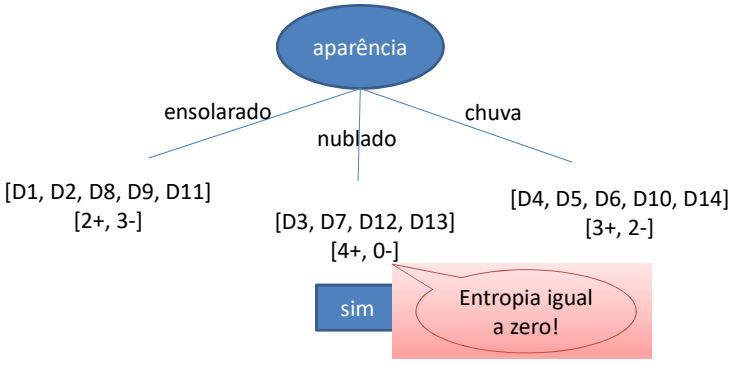
Construindo uma AD

- Qual deverá ser o nó raiz da árvore?
  - O nó raiz é identificado calculando o **ganho de informação** para cada atributo do conjunto de treinamento (menos o atributo classe).
- Calculando o Ganho para cada atributo, teremos:
  - Ganho(S, aparência) = 0.246
  - Ganho(S, umidade) = 0.151
  - Ganho(S, vento) = 0.048
  - Ganho(S, temperatura) = 0.029
    - De acordo com a medida de **ganho de informação**, o atributo **aparência** é o que melhor prediz do atributo classe (jogar_tênis) sobre o conjunto de treinamento.

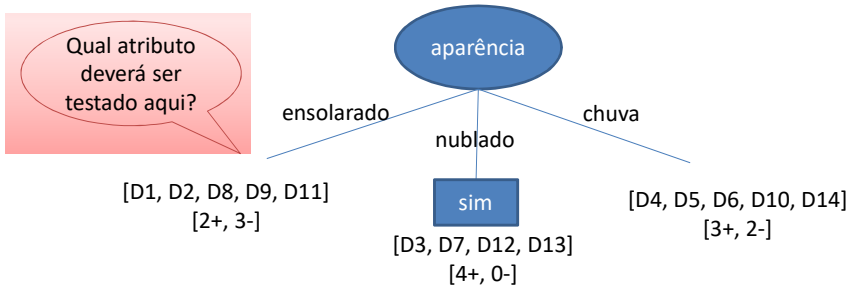
Construindo uma AD



Construindo uma AD



Construindo uma AD



Construindo uma AD

Dia	Aparência	Temperatura	Umidade	Vento	Jogar tênis
D1	ensolarado	quente	alta	fraco	não
D2	ensolarado	quente	alta	forte	não
D3	nublado	quente	alta	fraco	sim
D4	chuva	moderada	alta	fraco	sim
D5	chuva	fria	normal	fraco	sim
D6	chuva	fria	normal	forte	não
D7	nublado	fria	normal	forte	sim
D8	ensolarado	moderada	alta	fraco	não
D9	ensolarado	fria	normal	fraco	sim
D10	chuva	moderada	normal	fraco	sim
D11	ensolarado	moderada	normal	forte	sim
D12	nublado	moderada	alta	forte	sim
D13	nublado	quente	normal	fraco	sim
D14	chuva	moderada	alta	forte	não

Construindo uma AD

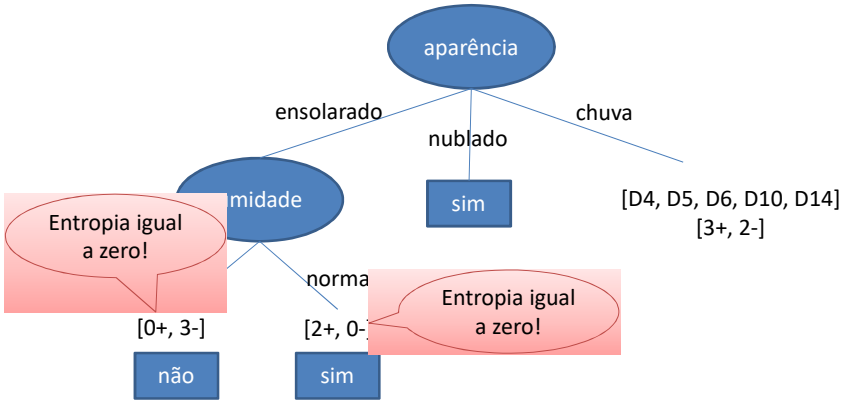
- $S_{ensolarado}$ = {D1, D2, D8, D9, D11}

Dia	Aparência	Temperatura	Umidade	Vento	Jogar tênis
D1	ensolarado	quente	alta	fraco	não
D2	ensolarado	quente	alta	forte	não
D8	ensolarado	moderada	alta	fraco	não
D9	ensolarado	fria	normal	fraco	sim
D11	ensolarado	moderada	normal	forte	sim

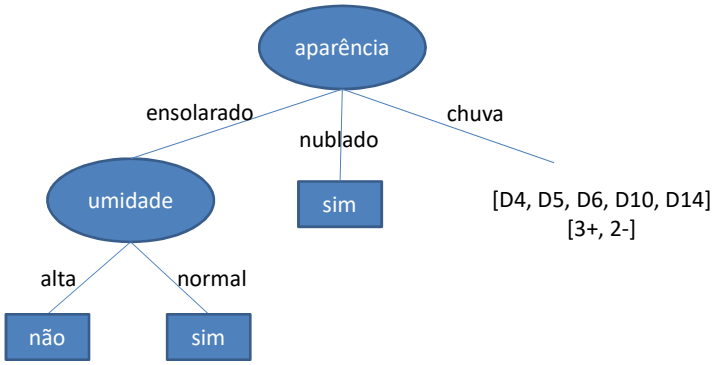
- Calculando os ganhos:
  - Ganho($S_{ensolarado}$, umidade) = 0.970 - (3/5)0.0 - (2/5)0.0 = 0.970
  - Ganho($S_{ensolarado}$, temperatura) = 0.970 - (2/5)1.0 - (2/5)0.0 - (1/5)0.0 = 0.570
  - Ganho($S_{ensolarado}$, vento) = 0.970 - (2/5)1.0 - (3/5)0.918 = 0.019

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

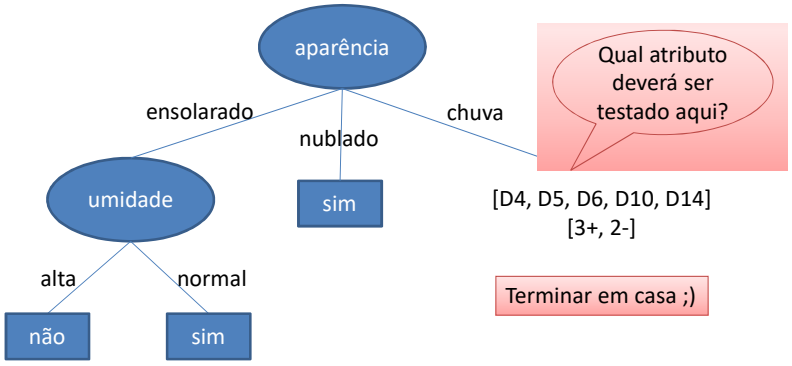
Construindo uma AD



Construindo uma AD



Construindo uma AD



Missing values

- Como lidar com atributos sem valores (ausentes ou perdidos por alguma razão)?
  - As instâncias com valores ausentes são ignoradas;
  - O valor "ausente" é usado como um valor comum, podendo ser um valor testado, inclusive;
  - Diante de um valor "ausente", segue-se por um outro ramo, por exemplo, o que for mais popular (ou seja, que tenha mais instâncias associadas a ele).
  - Permitir que a instância siga por vários ramos, sendo que é dado um peso para cada ramo em função do número de instâncias associados a ele; os resultados são recombinados (com os pesos) para se decidir pela classe.

Poda (*Pruning*)

- Maneira de tratar o ruído e o *overfitting* em árvores de decisão.
  - **Pré-poda**: durante a fase de geração da hipótese alguns exemplos de treinamento são deliberadamente ignorados.
  - **Pós-poda**: após a geração da hipótese, esta é generalizada, e algumas partes suas são eliminadas (poda de nós-folha)
    - Mais usada, pois se tem mais informação; no caso anterior, não se sabe onde parar a poda.
  - A folha resultante da poda deve ser marcada com a classe majoritária ou uma distribuição de probabilidade de cada classe.

Métodos de *Pruning*

- Métodos para determinar quais sub-árvores podar:
  - **Validação cruzada:** Reservar alguns dados de treinamento (conjunto de validação) para avaliar utilidade das sub-árvores.
  - **Teste estatístico:** Usa o teste para determinar se a regularidade observada foi devida ao acaso.
  - **Comprimento da descrição mínima (MDL):** Determina se a complexidade adicional da hipótese é mais ou menos complexa que lembrar explicitamente as exceções resultantes da poda.

Pseudocódigo de *Pruning*

```
(01) função PruningTree(árvore, exemplos): retorna uma árvore
(02) início
(03)   divida os dados de treinamento em dois conjuntos: "criação" e
                                                          "validação".
(04)   construa a árvore completa a partir do conjunto de "criação".
(05)   até que a precisão no conjunto de "validação" diminua faça:
(06)     para cada nó não-folha n na árvore faça
(07)       pode temporariamente a sub-árvore abaixo de n e a substitua
                              por uma folha rotulada com a classe
                                        mais frequente naquele nó.
(08)       avalie a árvore podada com o conjunto de validação e grave
                                                    a sua precisão.
(09)     pode permanentemente os nós que resultaram no maior ganho de
                                  precisão no conjunto de validação.
(10) fim.
```

Links interessantes:

- Softwares:
  - C4.5 (free)
  - See5.0 (limitado, para MS. Windows)
  - Weka 3 (http://www.cs.waikato.ac.nz/ml/weka/)

- Applet:

  http://www.cs.ualberta.ca/%7Eaixplore/learning/DecisionTrees/Applet/DecisionTreeApplet.html