Aprendizagem de Máquina

Alceu S. Britto Jr.

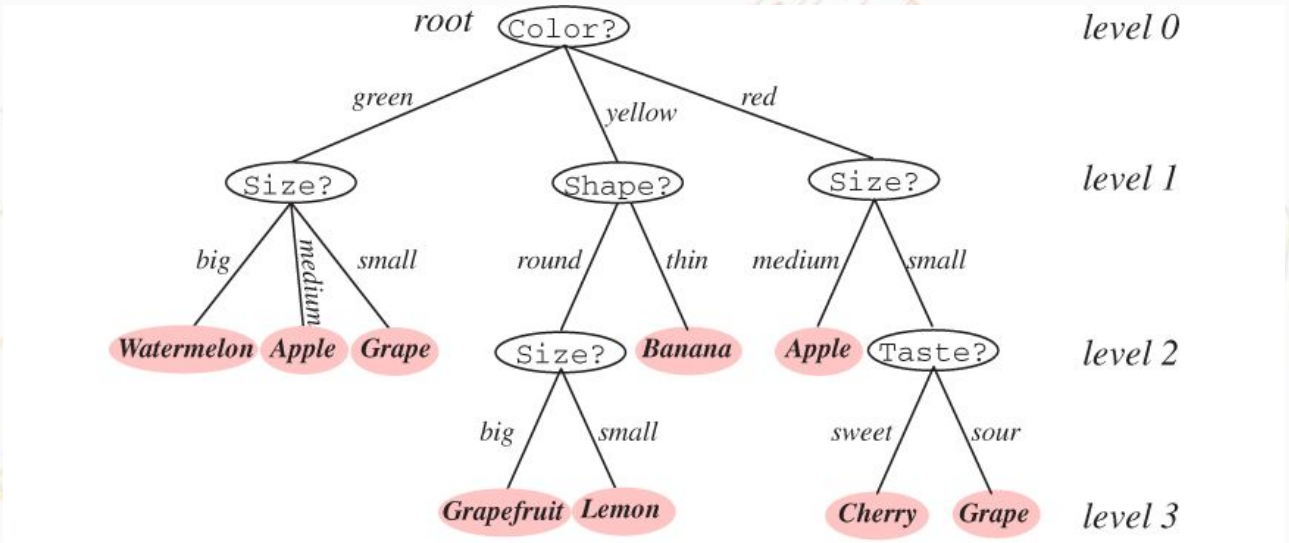
Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná (PUCPR)

ÁRVORES DE DECISÃO

Referências

- Duda R., Hart P., Stork D. *Pattern Classification 2ed*. Willey Interscience, 2002. Capítulo 8.
- Mitchell T. *Machine Learning*. WCB McGraw–Hill, 1997. Capítulo 3.
- Quinlan,J.R.: C4.5: Programs for Machine Learning. Morgan Kauffman, 1993

Exemplo



Classification in a basic decision tree proceeds from top to bottom. The questions asked at each node concern a particular property of the pattern, and the downward links correspond to the possible values. Successive nodes are visited until a terminal or leaf node is reached, where the category label is read. Note that the same question, `Size?`, appears in different places in the tree and that different questions can have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled by the same category (e.g., **Apple**).

Representação de Árvores de Decisão

- Uma instância é classificada inicialmente pelo **nó raiz**, testando o atributo especificado por este nó.
- Em seguida, movendo-se através do ramo correspondendo ao valor do atributo no exemplo dado.
- Este processo é repetido para a sub-árvore originada no novo nó.

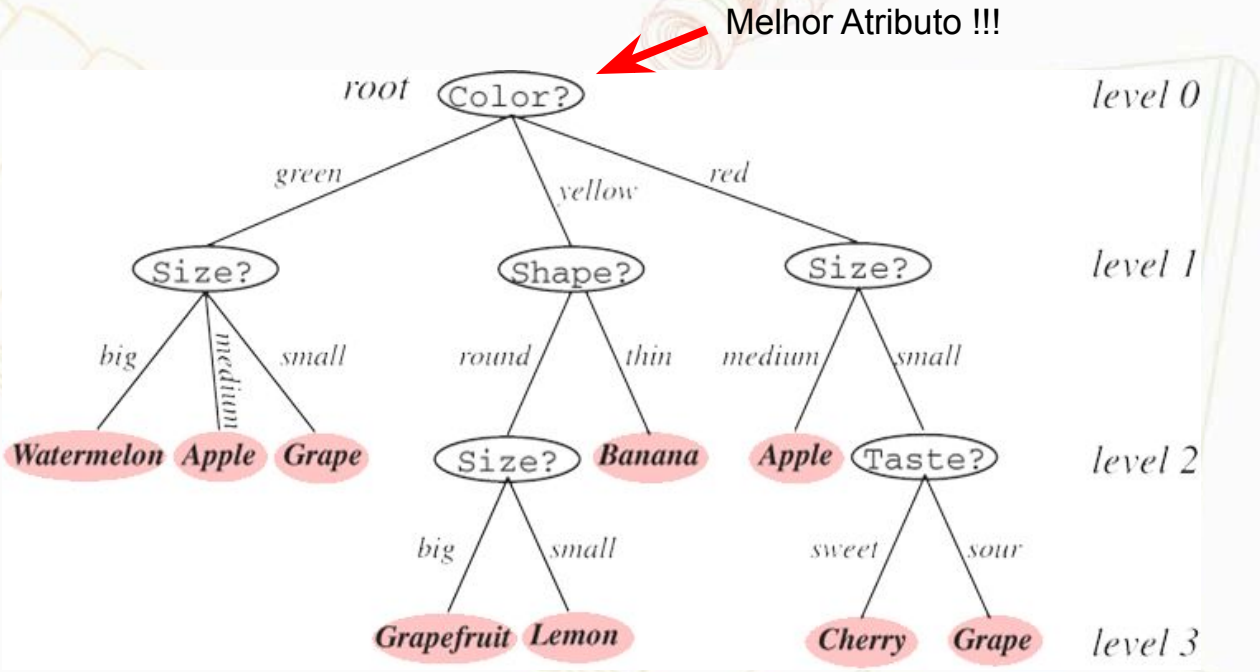
Algoritmo Básico para Aprendizagem de Árvores de Decisão

- **Algoritmo base**: ID3 e seu sucessor, o C4.5.
- O algoritmo ID3 “aprende” árvores de decisão construindo–as de cima para baixo, começando com a questão:

  “Qual atributo deve ser testado na raiz da árvore?”

- Para responder esta questão, cada atributo da instância é avaliado usando um teste estatístico para determinar quão bem ele sozinho classifica os exemplos de treinamento.

Exemplo



Algoritmo Básico para Aprendizagem de Árvores de Decisão

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A}))
- End
- Return *Root*

Algoritmo Básico para Aprendizagem de Árvores de Decisão

- **Escolha Central**: selecionar qual atributo testar em cada nó da árvore.

- Devemos selecionar:
  - Atributo que é mais útil para classificar os exemplos.
  - Como sabemos qual é o mais útil?

- Medida Quantitativa:
  - **Ganho de Informação** = mede quão bem um atributo separa os exemplos de treinamento de acordo com a classificação alvo.

Entropia

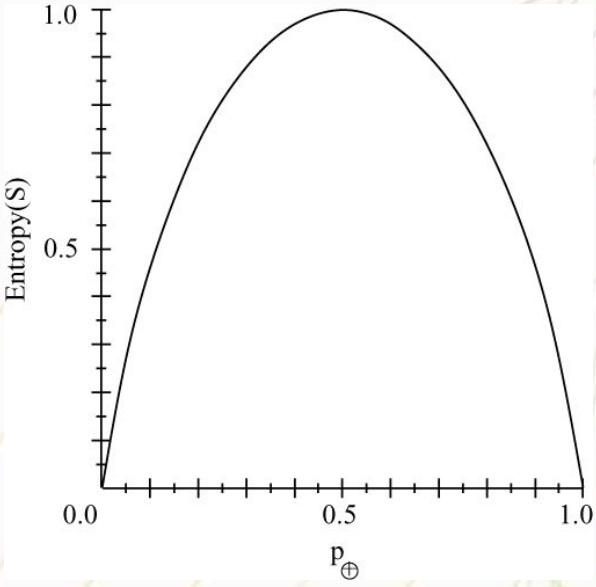
- Caracteriza a (im)pureza de uma coleção arbitrária de exemplos.

- Dado uma coleção *S* contendo exemplos positivos (+) e negativos (–) de algum conceito alvo, a entropia de *S* relativa a esta classificação booleana é:

$$Entropia(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

  - p_+ é a proporção de exemplos positivos em *S*
  - p_- é a proporção de exemplos negativos em *S*

Entropia



- A função entropia relativa a uma classificação booleana, como a proporção, p_+ de exemplos positivos varia entre 0 e 1.

Entropia

- Generalizando para o caso de um atributo alvo aceitar *c* diferentes valores, a entropia de *S* relativa a esta classificação *c*–classes é definida como:

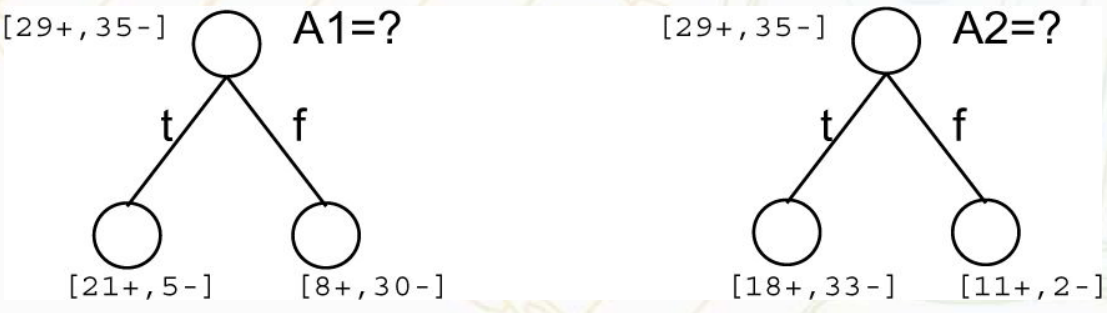
$$Entropia(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i$$

onde p_i é a proporção de *S* pertencendo a classe *i*.

Ganho de Informação

- *Gain* (*S,A*) = redução esperada na entropia devido a ordenação sobre *A*, ou seja, a redução esperada na entropia causada pela partição dos exemplos de acordo com este atributo *A*.

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



Exemplo Ilustrativo

- Atributo alvo: *PlayTennis* (*Yes, No*)

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Tarefa de Aprendizagem

- O atributo *PlayTennis* indica se eu jogo ou não tenis naquele dia.

Qual é a tarefa de aprendizagem ?

Aprender a predizer o valor de *PlayTennis* para um dia qualquer baseando–se apenas nos valores dos outros atributos (*Outlook, Temperature, Humidity, Wind*).

Exemplo Ilustrativo

- Primeiro passo: criação do nó superior da árvore de decisão.

Qual atributo deve ser testado primeiro na árvore?

- Determinar o ganho de informação (*Gain*) para cada atributo candidato (i.e. *Outlook*, *Temperature*, *Humidity* e *Wind*)

- Selecionar aquele cujo ganho de informação é o mais alto.

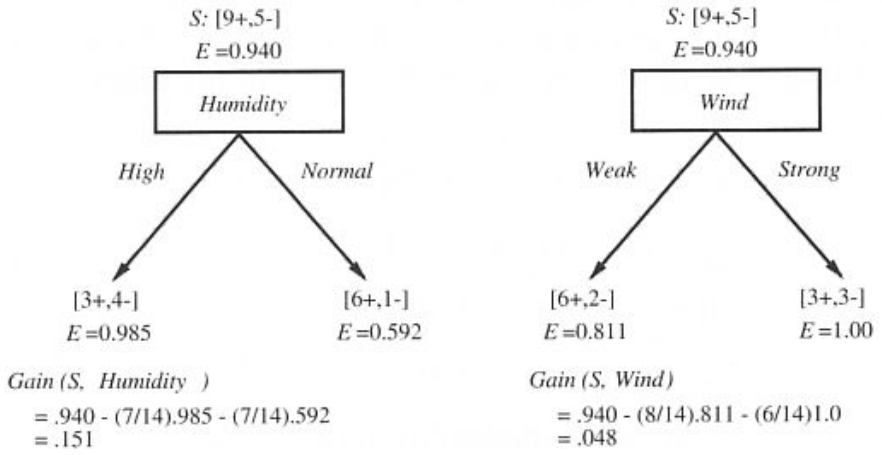
Entropia

- Exemplo: Sendo *S* uma coleção de 14 exemplos de algum conceito booleano, incluindo 9 exemplos positivos e 5 negativos [9+, 5–].

- A entropia de *S* relativa a classificação booleana é:

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

Exemplo Ilustrativo



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

FIGURE 3.3
Humidity provides greater information gain than *Wind*, relative to the target classification. Here, E stands for entropy and S for the original collection of examples. Given an initial collection S of 9 positive and 5 negative examples, $[9+, 5-]$, sorting these by their *Humidity* produces collections of $[3+, 4-]$ ($Humidity = High$) and $[6+, 1-]$ ($Humidity = Normal$). The information gained by this partitioning is .151, compared to a gain of only .048 for the attribute *Wind*.

Ganho de Informação

- S é uma coleção de (dias) exemplos de treinamento descritos por atributos incluindo *Wind*. Temos 14 exemplos.

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

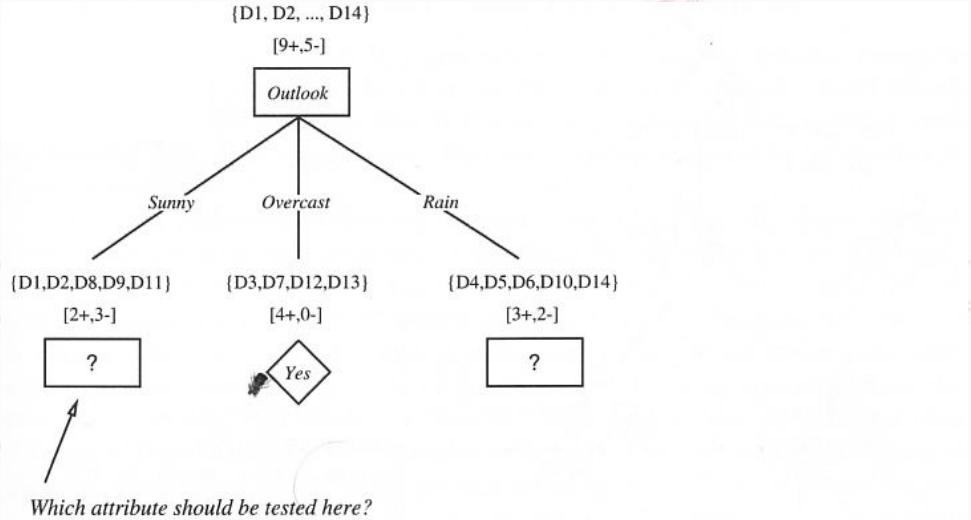
Exemplo Ilustrativo

Exemplo:

- *Gain* (*S, Outlook*) = 0.246
- *Gain*(*S, Humidity*) = 0.151
- *Gain* (*S, Wind*) = 0.048
- *Gain* (*S, Temperature*) = 0.029

- Ou seja, o atributo *Outlook* fornece a melhor predição do atributo alvo, *PlayTennis*, sobre os exemplos de treinamento (Fig 3.4)

Exemplo Ilustrativo



S_{sunny} = {D1,D2,D8,D9,D11}

$Gain\ (S_{sunny}, Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$

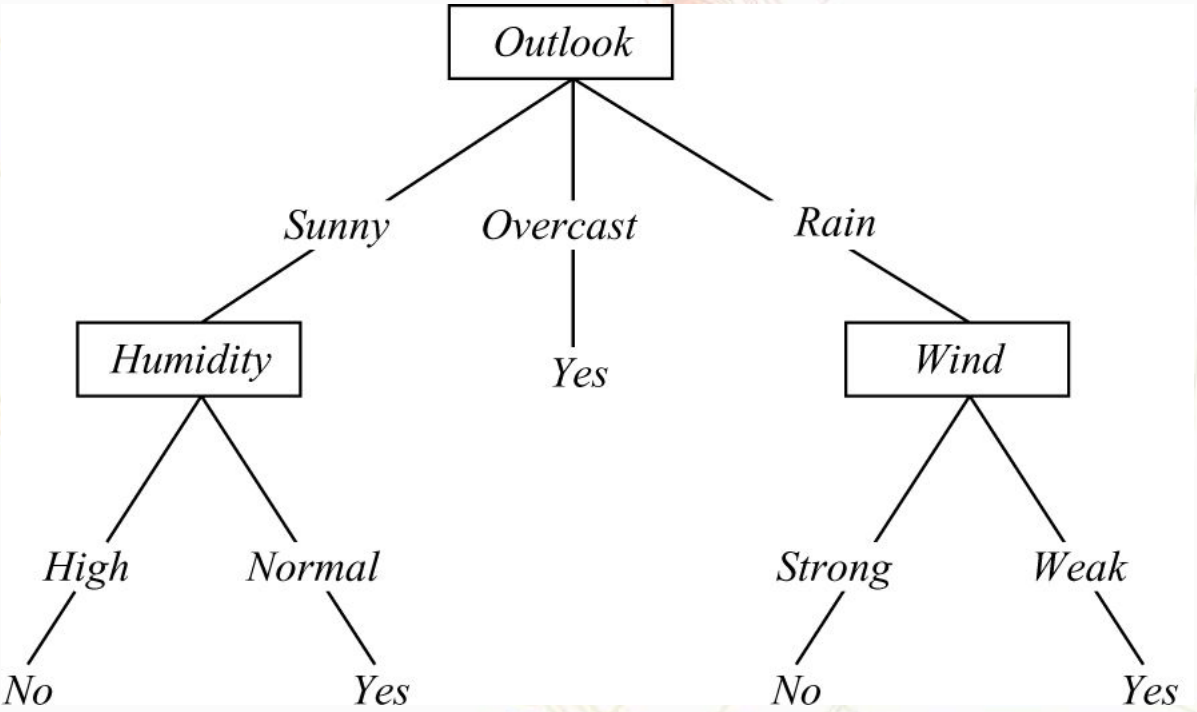
$Gain\ (S_{sunny}, Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$

$Gain\ (S_{sunny}, Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$

FIGURE 3.4
The partially learned decision tree resulting from the first step of ID3. The training examples are sorted to the corresponding descendant nodes. The *Overcast* descendant has only positive examples and therefore becomes a leaf node with classification *Yes*. The other two nodes will be further expanded, by selecting the attribute with highest information gain relative to the new subsets of examples.

Exemplo Ilustrativo

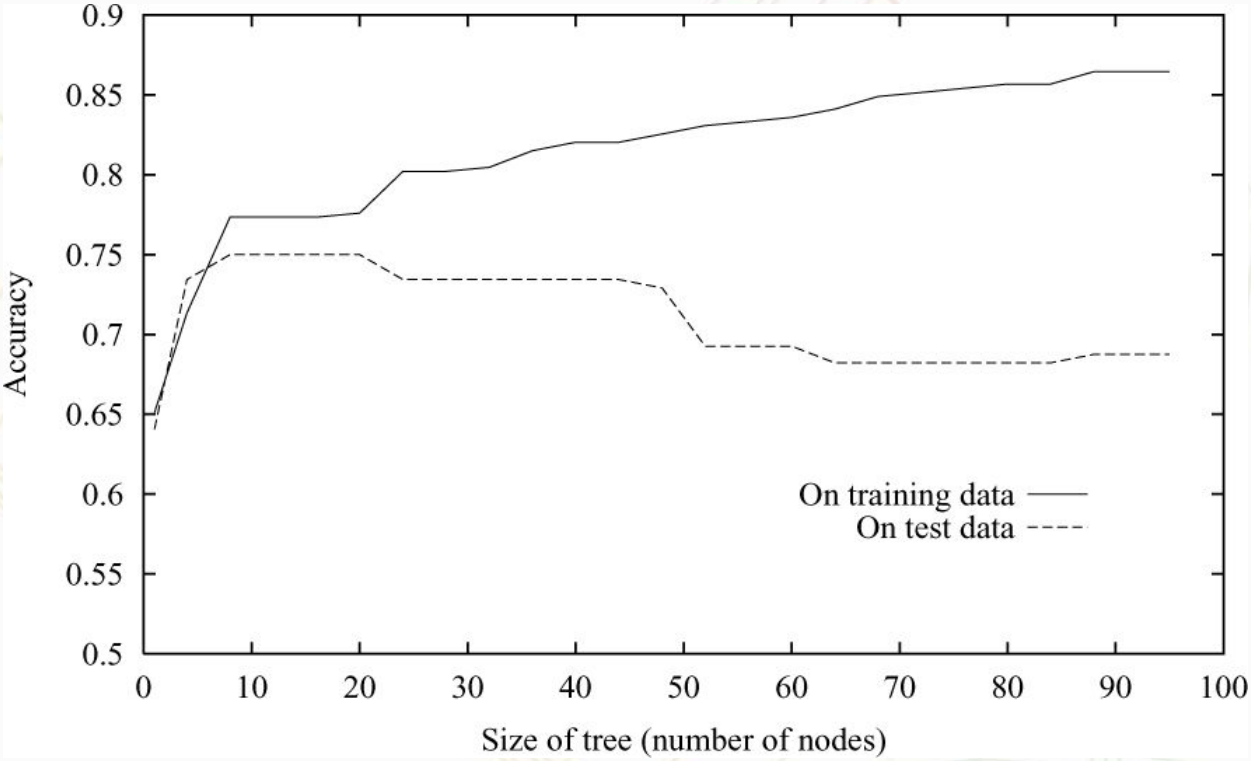
- Árvore de decisão final.



Árvore de Decisão para PlayTennis

- Representação de árvores de decisão:
  - Cada nó interno testa um atributo
  - Cada ramo corresponde ao valor do atributo
  - Cada folha atribui uma classificação

Sobreajuste no Treinamento de Árvores de Decisão



Evitando Sobreajuste

- *Podar um nó de decisão* → consiste em remover a sub-árvore enraizada naquele nó, tornando–o um nó folha.
- Atribuir a este nó, a classificação mais comum dos exemplos de treinamento afiliados com aquele nó.
- Nós são removidos somente se a árvore aparada resultante não apresenta um comportamento pior do que a original sobre o conjunto de validação

Atributos de Valor Continuo

Na definição da ID3 temos as restrições:

- 1. Atributo alvo deve ter valor discreto
- 2. Os atributos testados nos nós de decisão devem também ser de valor discreto.

A segunda restrição pode ser removida.

- Definir dinamicamente novos atributos de valor discreto que particionam o valor do atributo contínuo em um conjunto discreto de intervalos.

- *A* = atributo de valor contínuo → criar um novo atributo A_c que é verdadeiro se $A < c$ e falso caso contrário.

Como identificar o limiar *c* ???

- Valor que produza o maior ganho de informações.

- Árvores (Prós)

- Simplicidade para compreensão e interpretação
  - árvores de decisão são facilmente compreendidas após uma breve explicação
- Os dados não necessitam de pré-processamento
  - outras técnicas normalmente exigem normalização de dados.
- Lidam tanto com dados numéricos quanto categóricos
  - outras técnicas normalmente lidam somente com um único tipo de variável.

Árvores - (Prós)

- Emprega um modelo “caixa branca”
  - Se uma dada situação é observável em um modelo, a explicação para a condição é facilmente feita através da lógica booleana.
- Possibilidade de validar um modelo através de testes estatísticos.
  - é possível avaliar a confiabilidade do modelo.
- Robustez.
  - Bom desempenho mesmo se as suposições iniciais do modelo de dados forem violadas.
- Bom desempenho em grandes conjuntos de dados em um tempo curto.
  - Grandes quantidades de dados podem ser analisados utilizando recursos computacionais comuns.

Contras (Limitações)

- O problema de aprender uma árvore de decisão ótima é NP-Completo.
  - Os algoritmos práticos de aprendizagem de árvore de decisão são baseados em heurísticas (e.g. algoritmo guloso) onde decisões ótimas locais são tomadas em cada nó.
- O aprendizado de árvores de decisão pode criar árvores muito complexas que não generalizam bem os dados.
  - Sobreajuste (Overfitting!!). Mecanismos de poda são necessários para evitar este problema.

Contras (Limitações)

- Alguns conceitos são difíceis de serem aprendidos, pois, árvores de decisão não expressa-os facilmente.
  - Problemas XOR, paridade e multiplexador
  - Nestes casos as árvores de decisão se tornam proibitivamente grandes.
- Para dados com variáveis categóricas, com diferentes níveis, o ganho de informação é tendencioso em favor daqueles atributos que possuem mais níveis.