Árvores de Decisão

Sistemas Inteligentes

Uma Abordagem típica em aprendizagem simbólica

- Árvores de decisão: inductive decision trees (ID3)
  - Instâncias (exemplos) são representadas por pares atributo-valor
  - Fáceis de serem implementadas e utilizadas
  - aprendizagem não incremental
  - estatística (admite exceções)

Árvores de Decisão

- Uma árvore de decisão utiliza uma estratégia de *dividir-para-conquista*r:
  - Um problema complexo é decomposto em sub-problemas mais simples.
  - Recursivamente a mesma estratégia é aplicada a cada sub-problema.
- A capacidade de discriminação de uma árvore vem da:
  - Divisão do espaço definido pelos atributos em sub-espaços.
  - A cada sub-espaço é associada uma classe.

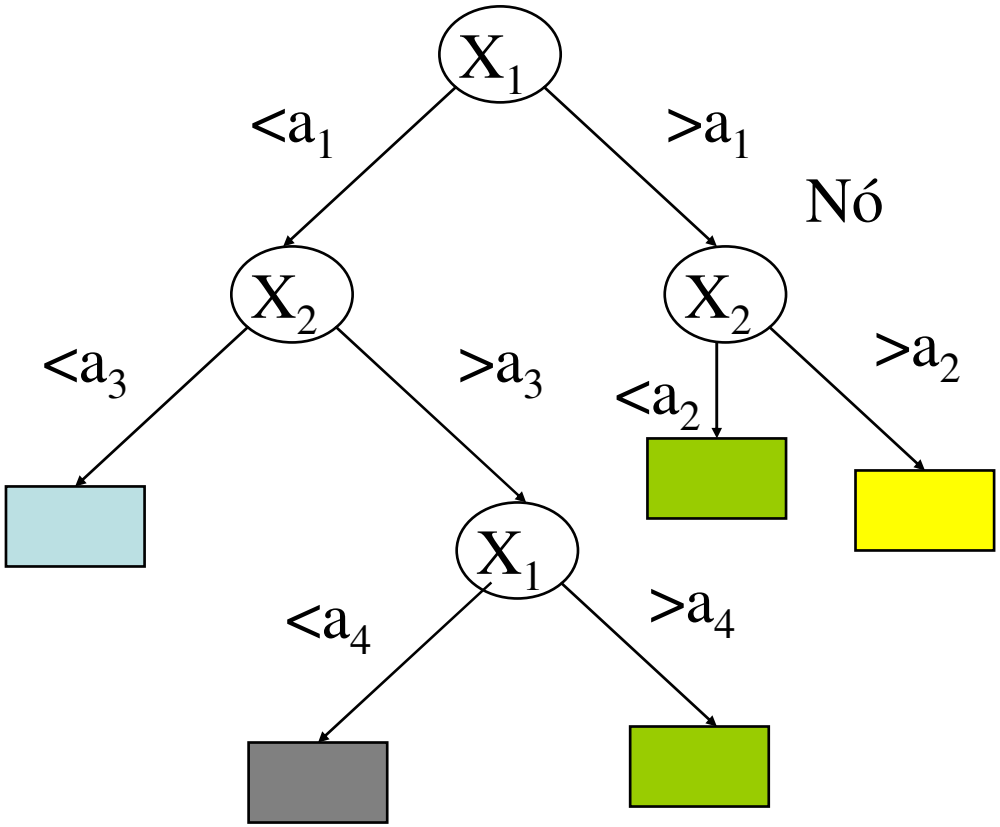
Aprendizagem indutiva

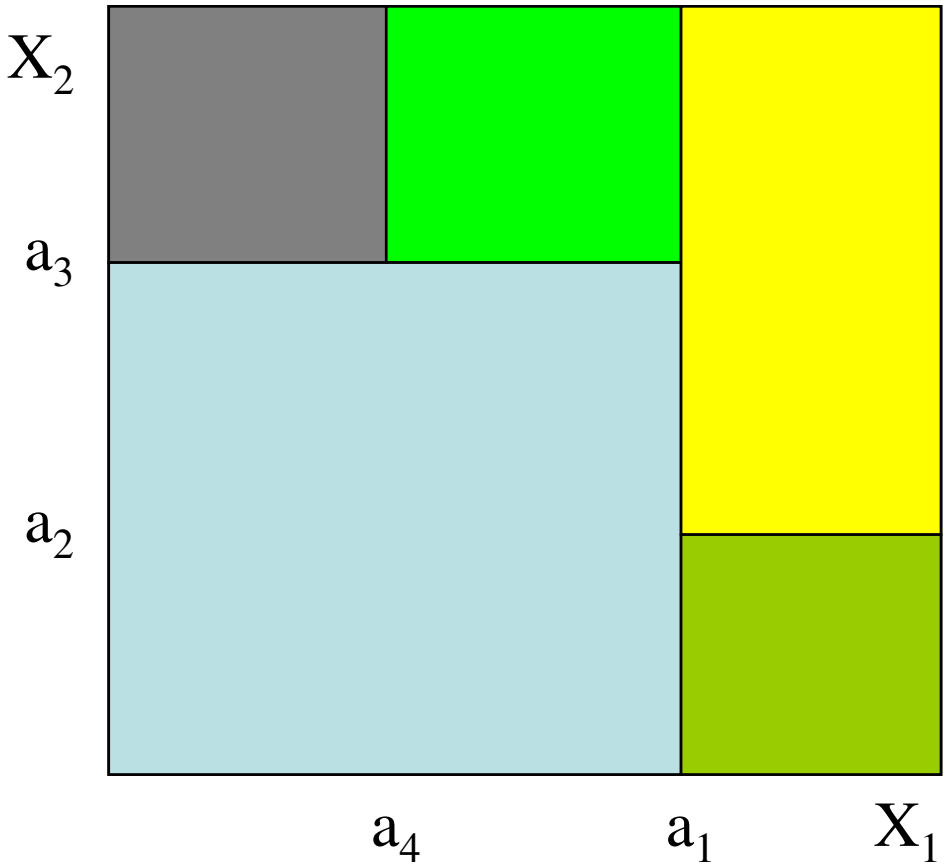
- Pode ser
  - **incremental**: atualiza hipótese a cada novo exemplo
    - mais flexível, situada... Porém a ordem de apresentação é importante (backtracking)
  - **não incremental**: gerada a partir de todo conjunto de exemplos
    - mais eficiente e prática

Árvores de Decisão

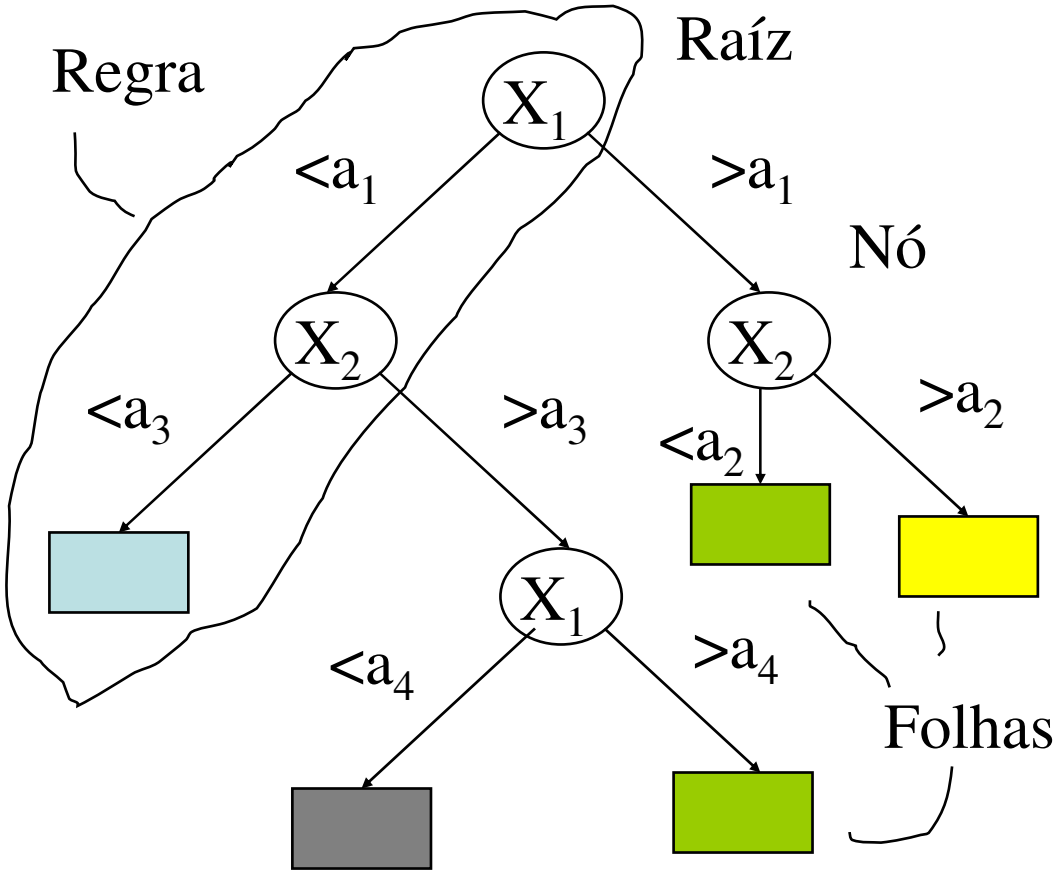
- Crescente interesse
  - CART (Breiman, Friedman, et.al.)
  - C4.5 (Quinlan)
  - S plus , Statistica, SPSS, SAS

Árvores de Decisão





O que é uma Árvore de Decisão



• Representação por árvores de decisão:

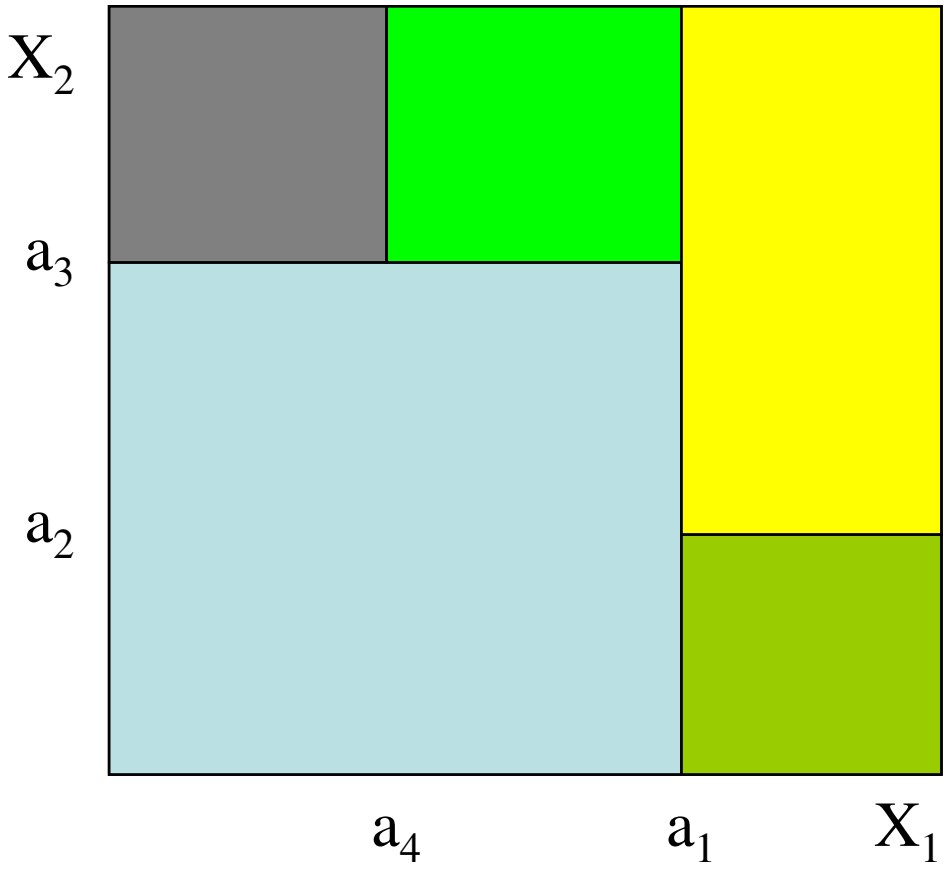
– Cada nó de decisão contem um teste num atributo.

– Cada ramo descendente corresponde a um possível valor deste atributo.

– Cada Folha está associada a uma classe.

– Cada percurso na árvore (da raiz à folha) corresponde a uma regra de classificação.

Árvores de Decisão



- No espaço definido pelos atributos:
  - Cada folha corresponde a uma região: Hiper-retângulo
  - A intersecção dos hiper - retângulos é vazia
  - A união dos hiper-retângulos é o espaço completo

Quando usar árvores de decisão?

- Instâncias (exemplos) são representadas por pares atributo-valor
- Função objetivo assume apenas valores discretos
- Hipóteses disjuntivas podem ser necessárias
- Conjunto de treinamento possivelmente corrompido por ruído
- Exemplos:
  - Diagnóstico médico, diagnóstico de equipamentos, análise de crédito

Construção de uma Árvore de Decisão

- A idéia *base*:

  1. Escolher um atributo.

  2. Estender a árvore adicionando um ramo para cada valor do atributo.

  3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido)

  4. Para cada folha

     1. Se todos os exemplos são da mesma classe, associar essa classe à folha

     2. Senão repetir os passos 1 a 4

Exemplo

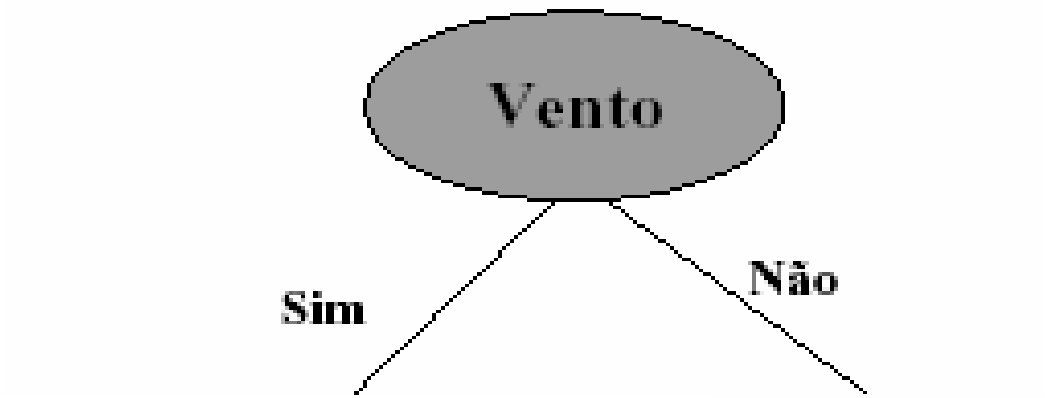
O conjunto de dados original

Tempo	Temperatura	Humidade	vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Nublado	81	75	Não	Sim
Chuva	71	91	Sim	Não

Exemplo

Seleciona um atributo

Qual o melhor atributo?



Tempo	Temperatura	Humidade	vento	Joga
Sol	80	90	Sim	Não
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Chuva	71	91	Sim	Não

Tempo	Temperatura	Humidade	vento	Joga
Sol	85	85	Não	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Nublado	81	75	Não	Sim

Critérios para Escolha do Atributo

- Como medir a *habilidade* de um dado atributo discriminar as classes?
- Existem muitas medidas.

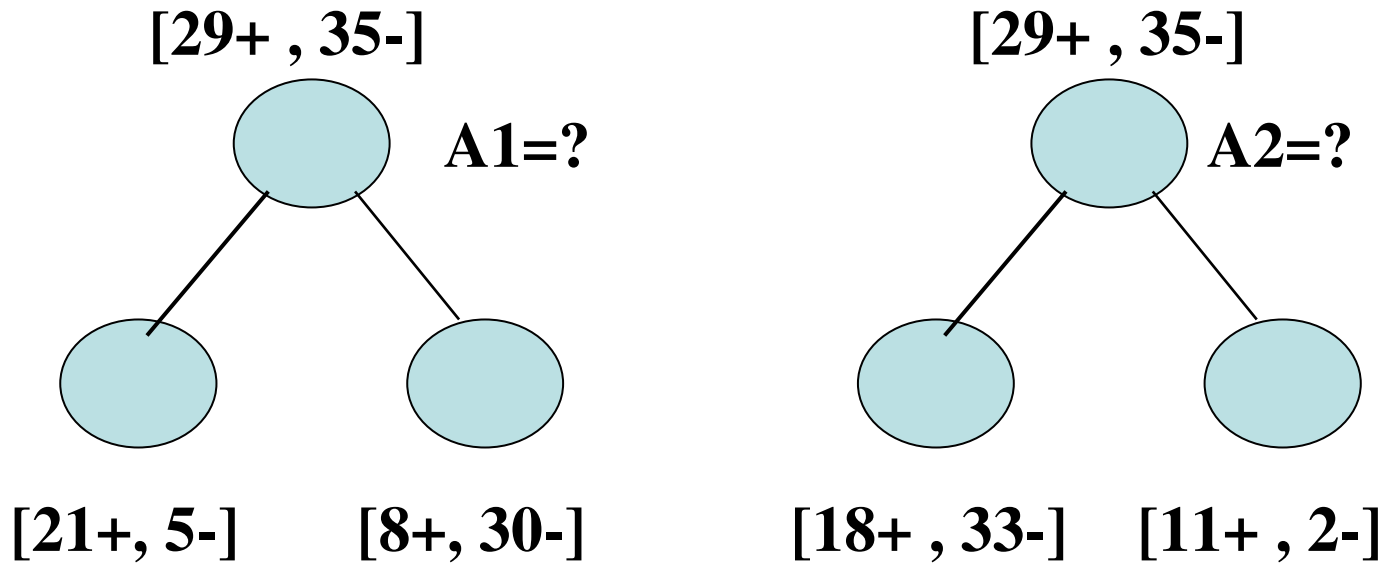
Todas concordam em dois pontos:

– Uma divisão que mantêm as proporções de classes em todas as partições é inútil.

– Uma divisão onde em cada partição todos os exemplos são da mesma classe tem utilidade máxima.

Critérios para Escolha do Atributo

- Qual é o melhor atributo?



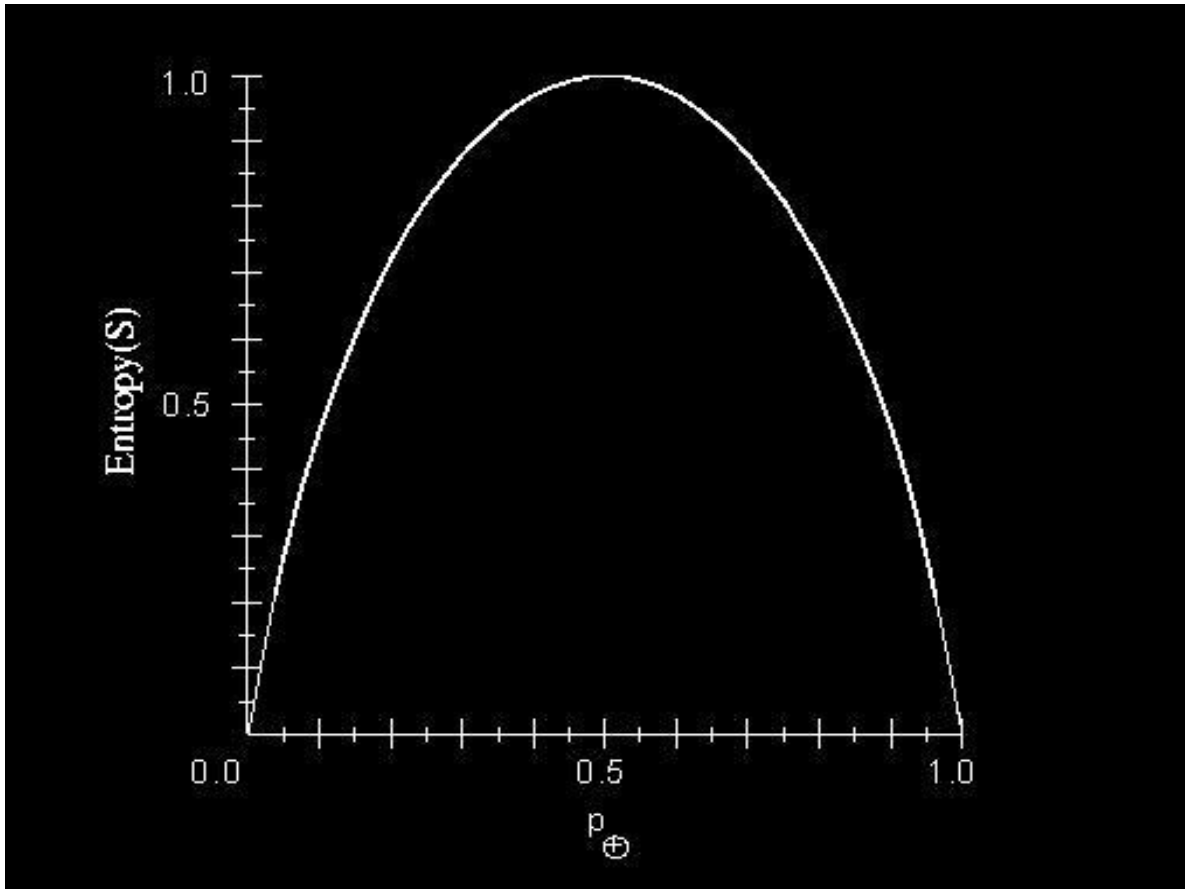
Entropia

- S é uma amostra dos exemplos de treinamento
- $p_{\oplus}$ é a proporção de exemplos positivos em S
- $p_{\ominus}$ é a proporção de exemplos negativos em S
- Entropia mede a "impureza" de S:
  - $Entropia(S) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$

Entropia - Exemplo I

- Se $p_{\oplus}$ é 1, o destinatário sabe que o exemplo selecionado será positivo
  - Nenhuma mensagem precisa ser enviada
  - Entropia é 0 (mínima)
- Se $p_{\oplus}$ é 0.5, um bit é necessário para indicar se o exemplo selecionado é $\oplus$ ou $\ominus$
  - Entropia é 1 (máxima)

Entropia - Gráfico



Entropia

- Entropia é uma medida da aleatoriedade

(impureza) de uma variável.

- A entropia de uma variável nominal X que pode tomar i valores:

$$entropia(X) = -\sum_i p_i \log_2 p_i$$

- A entropia tem máximo (log2 i) se $p_i = p_j$ para qualquer $i \neq j$
- A entropia(x) = 0 se existe um i tal que $p_i = 1$
- É assumido que 0 * log2 0 = 0

Entropia - Exemplo II

- Suponha que *S* é uma coleção de 14 exemplos, incluindo 9 positivos e 5 negativos
  - Notação: [9+,5-]
- A entropia de S em relação a esta classificação booleana é dada por:

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

Ganho de Informação

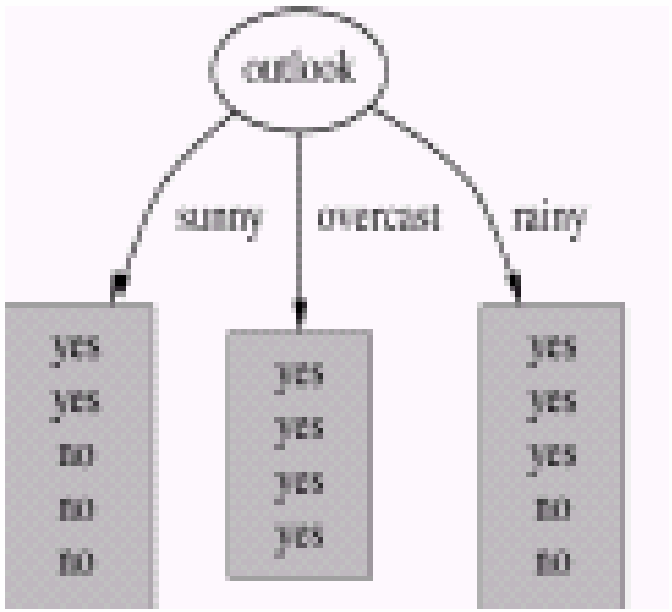
- No contexto das árvores de decisão a entropia é usada para estimar a aleatoriedade da variável a prever (classe).
- Dado um conjunto de exemplos, que atributo escolher para teste?
  - Os valores de um atributo definem partições do conjunto de exemplos.
  - O ganho de informação mede a redução da entropia causada pela partição dos exemplos de acordo com os valores do atributo.

Ganho de Informação

$$ganho(Exs, Atri) = entropia(Exs) - \sum_{v} \frac{\#Exs_v}{\#Exs} entropia(Exs_v)$$

A construção de uma árvore de decisão é guiada pelo objetivo de diminuir a entropia ou seja a aleatoriedade - dificuldade de previsão- da variável que define as classes

Cálculo do Ganho de Informação de um Atributo Nominal



- Informação da Classe:
  - p(sim) = 9/14
  - p(não) = 5/14
  - Ent(joga) = - 9/14 $\log_2$ 9/14
    - 5/14 $\log_2$ 5/14 = 0.940
- Informação nas partições:
  - p(sim | tempo=sol) = 2/5
  - p(não | tempo=sol) = 3/5

Cálculo do Ganho de Informação de um Atributo Nominal

	Sol	Nublado	Chuva
Sim	2	4	3
Não	3	0	2

- Informação nas partições:
  - Ent(joga | tempo=sol)
  - = -2/5 $\log_2$ 2/5 –3/5 $\log_2$ 3/5 = 0.971
  - Ent(joga | tempo=nublado) = 0.0
  - Ent(joga | tempo=chuva) = 0.971
  - Info(tempo) = 5/14*0.971 + 4/14*0+5/14*0.971= 0.693
- Ganho de Informação obtida neste atributo:
  - Ganho(tempo) = Ent(joga)-Info(tempo)
  - Ganho(tempo) = 0.940 – 0.693 = 0.247

Cálculo do Ganho para Atributos Numéricos

- Um teste num atributo numérico produz uma partição binária do conjunto de exemplos:
  - Exemplos onde valor_do_atributo < ponto_referência
  - Exemplos onde valor_do_atributo > ponto_referência
- Escolha do ponto de referência:
  - Ordenar os exemplos por ordem crescente dos valores do atributo numérico.
  - Qualquer ponto intermediário entre dois valores diferentes e consecutivos dos valores observados no conjunto de treinamento pode ser utilizado como possível ponto de referência

Cálculo do Ganho para Atributos Numéricos

Temperatu.	Joga
64	Sim
65	Não
68	Sim
69	Sim
70	Sim
71	Não
72	Não
72	Sim
75	Sim
75	Sim
80	Não
81	Sim
83	Sim
85	Não

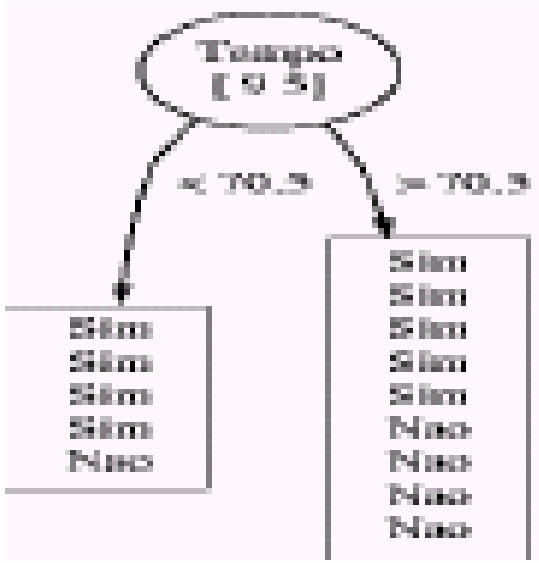
- Considere o ponto de referência temperatura = 70.5
- Um teste usando este ponto de referência divide os exemplos em duas classes:
  - Exemplos onde temperatura < 70.5
  - Exemplos onde temperatura > 70.5
- Como medir o ganho de informação desta partição?

Cálculo do Ganho para Atributos Numéricos

- É usual considerar o valor médio entre dois valores diferentes e consecutivos
- Fayyard e Irani (1993) mostram que de todos os possíveis pontos de referência aqueles que maximizam o ganho de informação separam dois exemplos de classes diferentes

Cálculo do Ganho para Atributos Numéricos

- Como medir o ganho de informação desta partição?
- Informação nas partições
  - p(sim | temperatura<70.5)=4/5
  - p(não | temperatura<70.5)=1/5
  - p(sim | temperatura>70.5)=5/9
  - p(não | temperatura>70.5)=4/9



Cálculo do Ganho para Atributos Numéricos

- Info(joga | temperatur<70.5) = -4/5 $\log_2$ 4/5 – 1/5 $\log_2$ 1/5 = 0.721
- Info(joga | temperatura >70.5) = -5/9 $\log_2$ 5/9 – 4/9 $\log_2$ 4/9 = 0.991
- Info(temperatura) = 5/14*0.721+9/14*0.991 = 0.895
- Ganho(temperatura) = 0.940 – 0.895 = 0.045 bits

Critérios de Parada

- Quando parar a divisão dos exemplos?
  - Todos os exemplos pertencem a mesma classe.
  - Todos os exemplos têm os mesmos valores dos atributos (mas diferentes classes).
  - O número de exemplos é inferior a um certo limite.
  - O mérito de todos os possíveis testes de partição dos exemplos é muito baixo.

Construção de uma Árvore de Decisão

- Input: Um conjunto exemplos
- Output: Uma árvore de decisão
- Função Geraarvore(Exs)
  - Se criterio_parada(Exs) = TRUE: retorna Folha
  - Escolhe o atributo que maximiza o critério_divisão(Exs)
  - Para cada partição *i* dos exemplos baseada no atributo escolhido: árvorei = Geraárvore(Exsi)
  - Retorna um nó de decisão baseado no atributo escolhido e com descendentes $árvore_i$.
  - Fim

Construção de uma Árvore de Decisão

- O problema de construir uma árvore de decisão:
  - Consistente com um conjunto de exemplos
  - Com o menor número de nós
  - É um problema *NP* completo.
- Dois problemas:
  - Que atributo selecionar para teste num nó?
  - Quando parar a divisão dos exemplos ?

Construção de uma Árvore de Decisão

- Os algoritmos mais populares:
  - Utilizam heurísticas que tomam decisões olhando para a frente um passo.
  - Não reconsideram as opções tomadas
    - Não há backtracking
    - Mínimo local

Construção de uma Árvore de Decisão

Exemplos de Treino

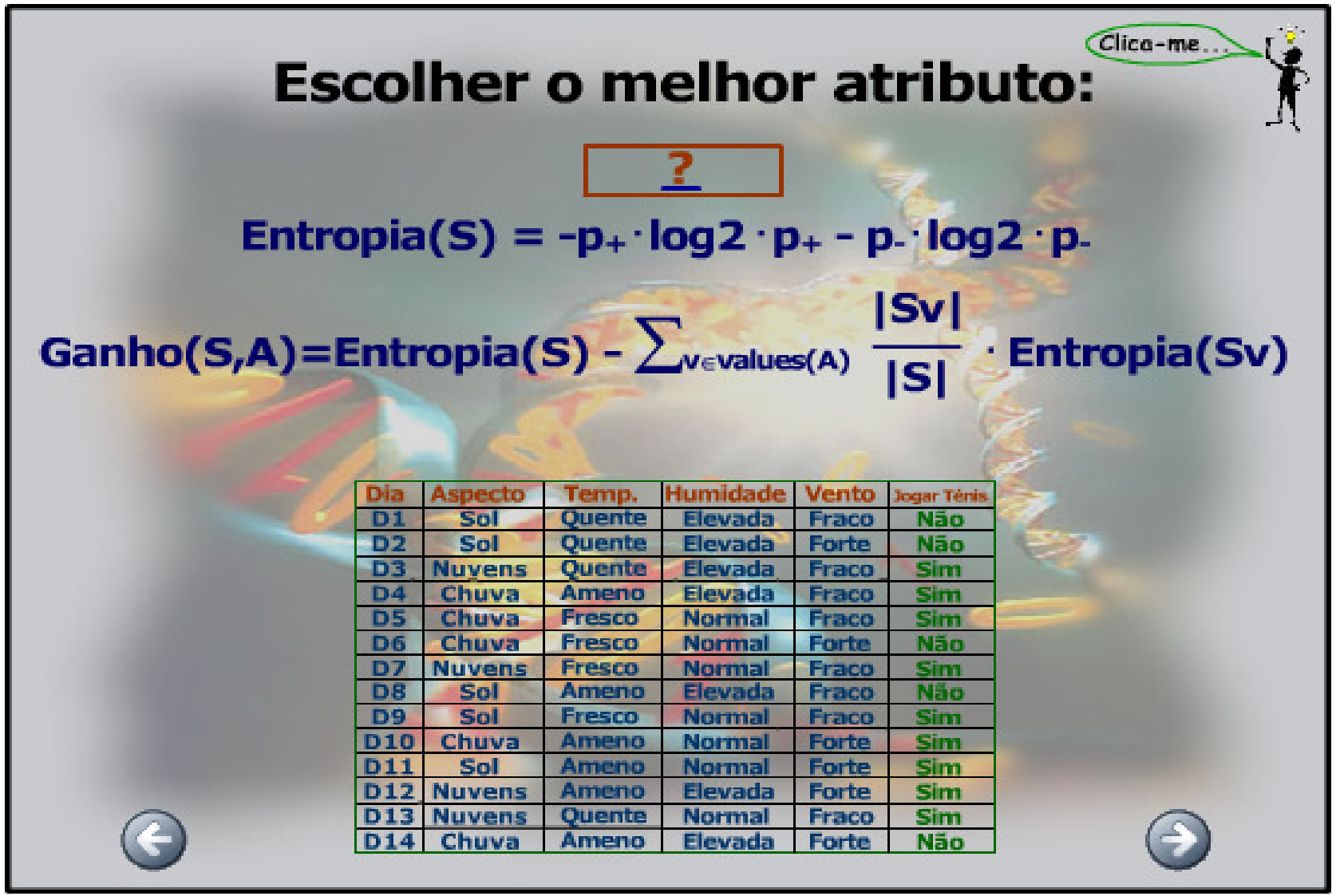
Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Referencia : http://bioinformatics.ath.cx

Construção de uma Árvore de Decisão

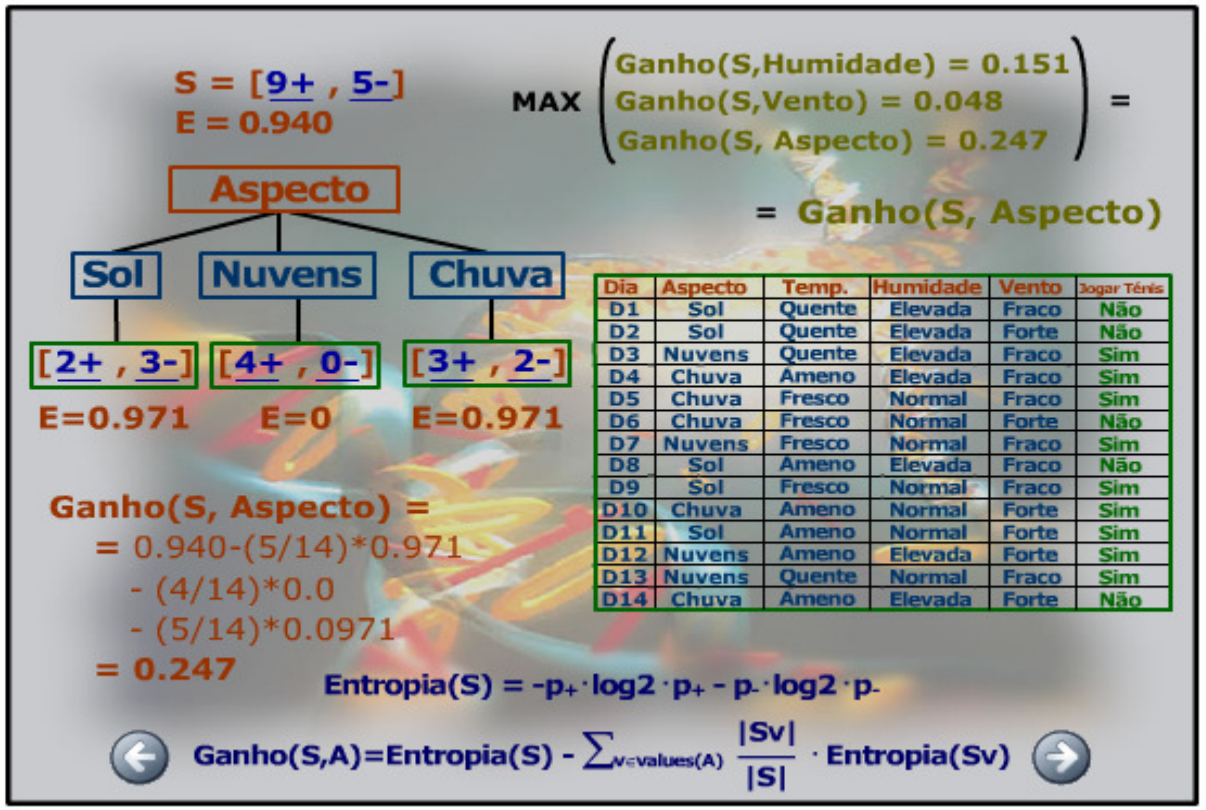
Construção de uma Árvore de Decisão

Construção de uma Árvore de Decisão



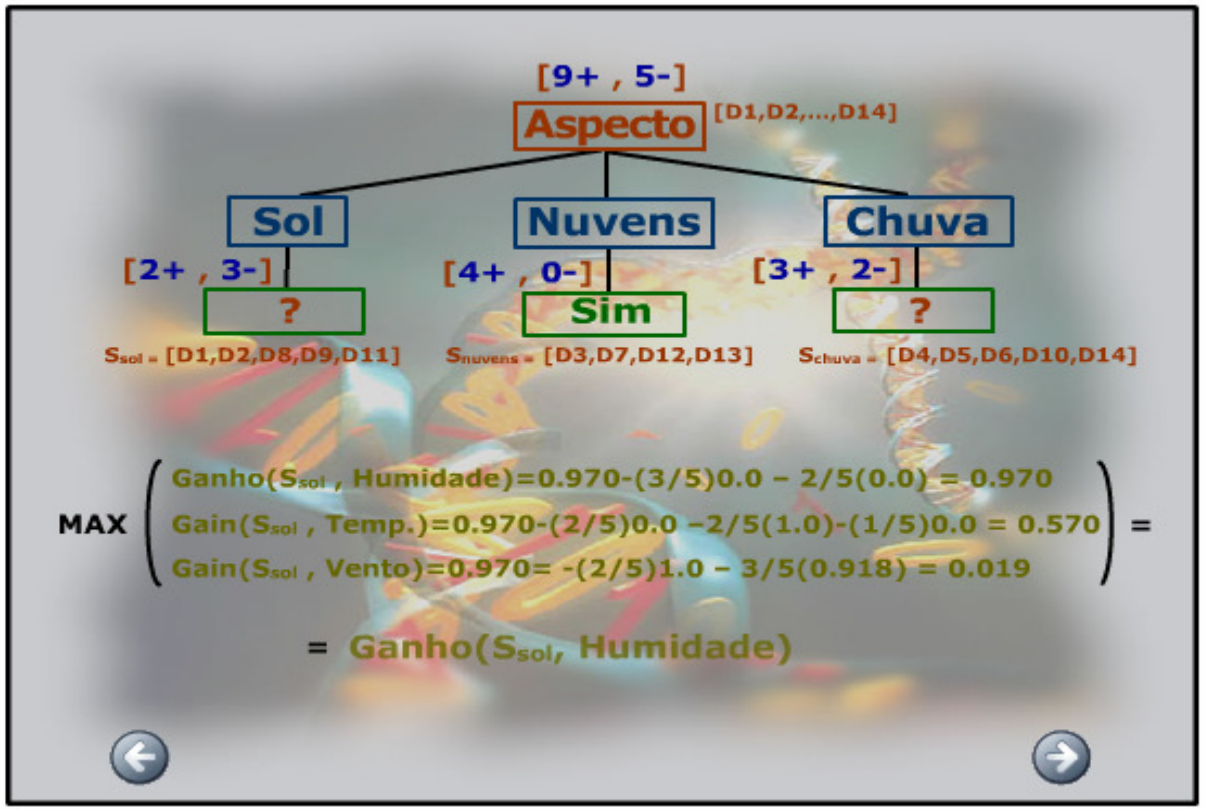
Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Construção de uma Árvore de Decisão



Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Construção de uma Árvore de Decisão

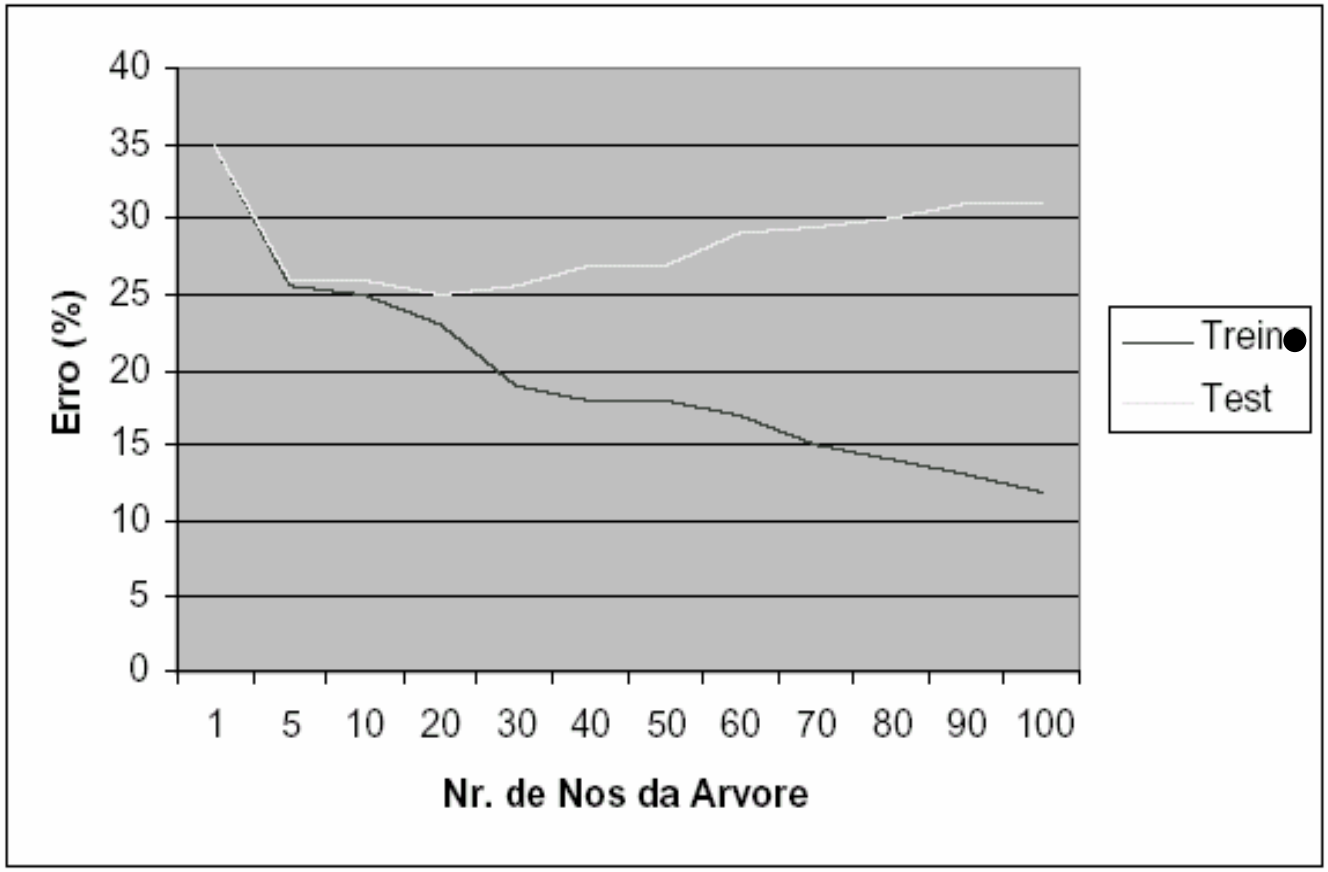


Construção de uma Árvore de Decisão

Sobre-ajustamento (Overfitting)

- O algoritmo de partição recursiva do conjunto de dados gera estruturas que podem obter um ajuste aos exemplos de treinamento perfeito.
  - Em domínios sem ruído o nr. de erros no conjunto de treinamento pode ser 0.
- Em problemas com *ruído* esta capacidade é problemática:
  - A partir de uma certa profundidade as decisões tomadas são baseadas em pequenos conjuntos de exemplos.
  - A capacidade de generalização para exemplos não utilizados no crescimento da árvore diminui.

Variação do erro com o no. de nós



Sobre-ajustamento (“*overfitting*”)

- Definição:
  - Uma árvore de decisão *d* faz sobre-ajustamento aos dados se existir uma árvore *d´* tal que:

  d tem menor erro que d´ no conjunto de treinamento

  mas *d´* tem menor erro na população.
- Como pode acontecer:
  - Ruído nos dados;
- O número de parâmetros de uma árvore de decisão cresce linearmente com o número de exemplos.
  - Uma árvore de decisão pode obter um ajuste perfeito aos dados de treinamento.

Sobre-ajustamento ("*overfitting*")

- Occam's razor: preferência pela hipótese mais simples.
  - Existem menos hipóteses simples do que complexas.
  - Se uma hipótese simples explica os dados é pouco provável que seja uma coincidência.
  - Uma hipótese complexa pode explicar os dados apenas por coincidência.

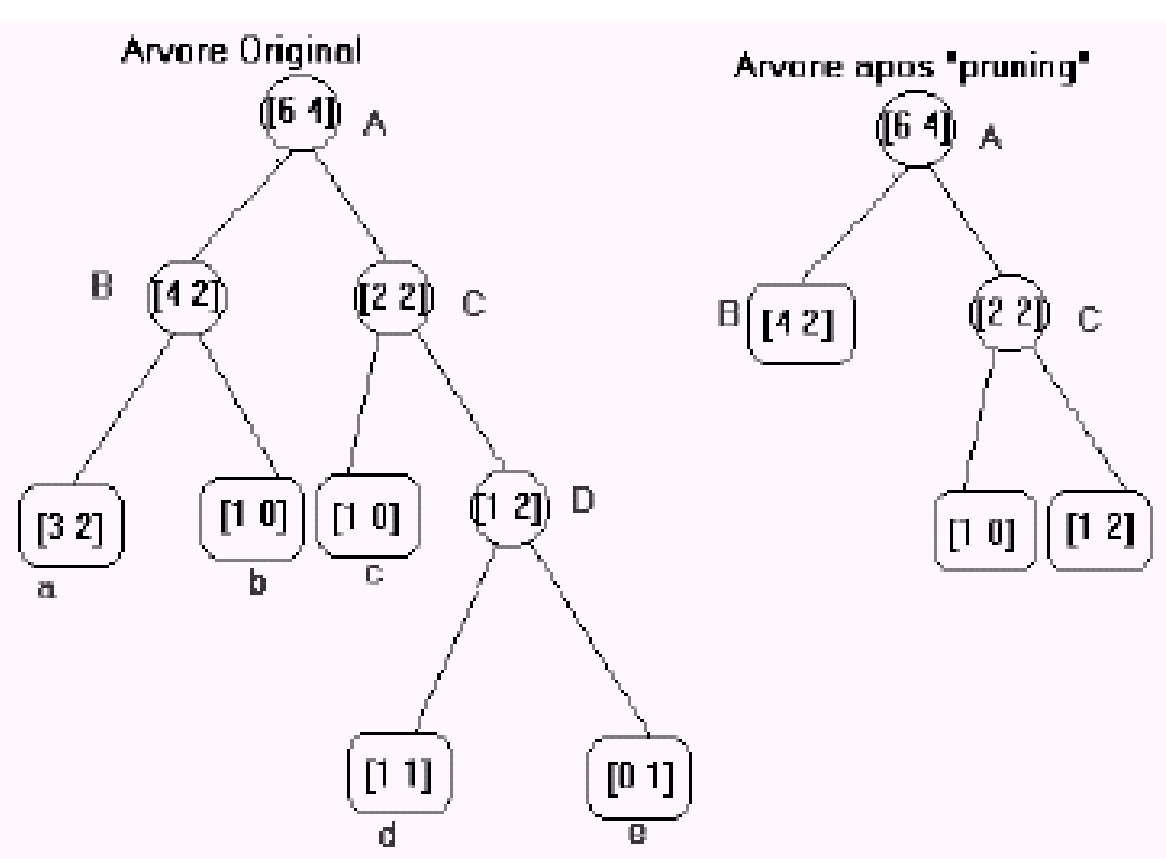
Simplificar a árvore

- Duas possibilidades:
  - Parar o crescimento da árvore mais cedo (pre-pruning).
  - Costruir uma árvore completa e podar a árvore (pos-pruning).
  - "*Growing and pruning is slower but more reliable*"
    - Quinlan, 1988

Um algoritmo básico de pruning

- Percorre a árvore em profundidade
- Para cada nó de decisão calcula:
  - Erro no nó
  - Soma dos erros nos nós descendentes
- Se o erro no nó é menor ou igual à soma dos erros dos nós descendentes o nó é transformado em folha.

Um algoritmo Básico de *Pruning*

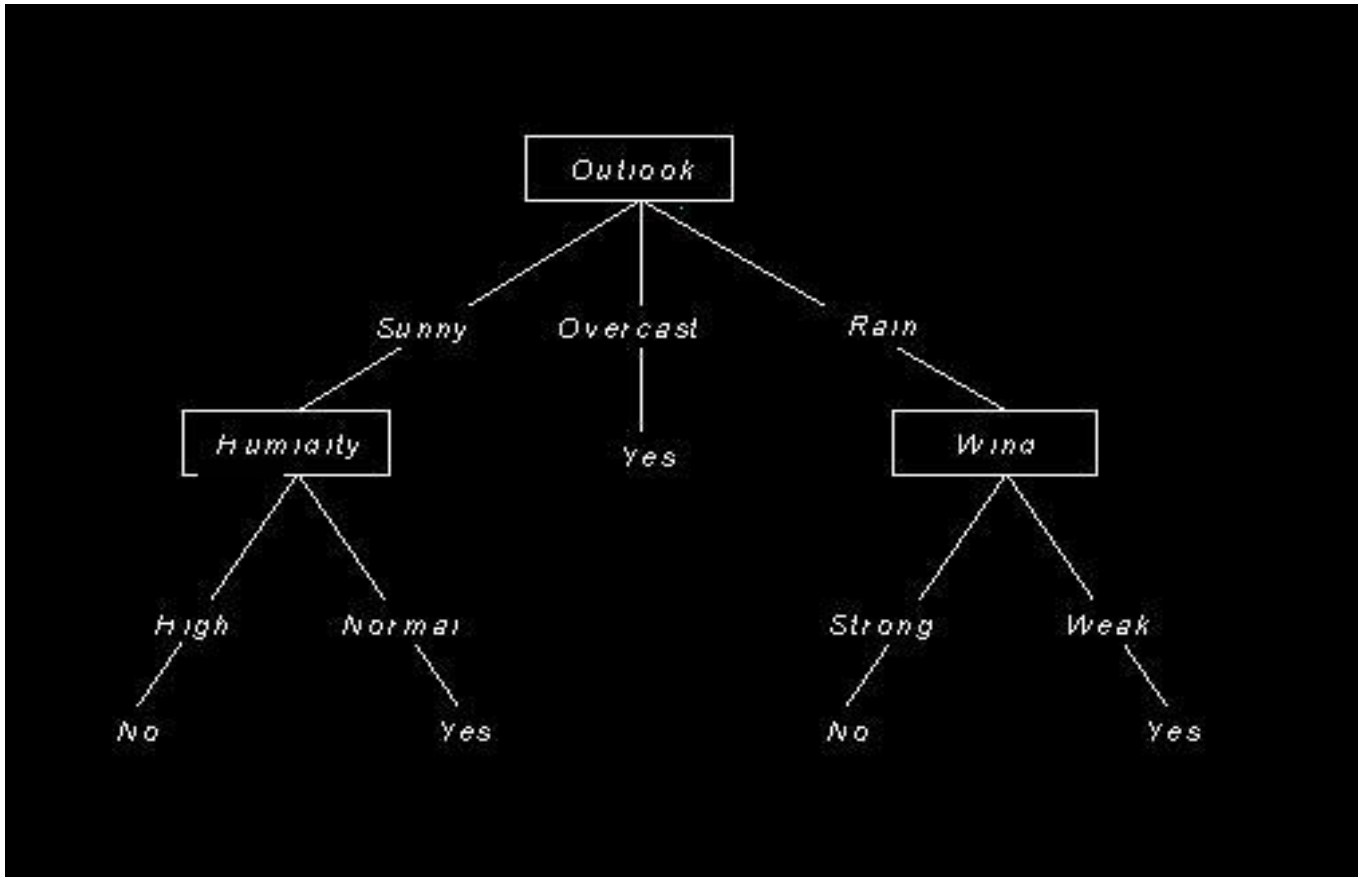


- Exemplo do nó B:
  - Erro no nó = 2
  - Soma dos erros nos nós descendentes:
    - 2 + 0
  - Transforma o nó em folha
    - Elimina os nós descendentes.

Critérios de como escolher a melhor árvore

- Obter estimativas confiáveis do erro a partir do conjunto de treinamento.
- Otimizar o erro num conjunto de validação independente do utilizado para construir a árvore.
- Minimizar:
  - *erro no treinamento* + *dimensão da árvore*
    - *Cost Complexity pruning (Cart)*
  - dimensão da árvore + quantidade de exemplos mal classificados
    - MDL pruning (Quinlan)

Convertendo uma árvore em regras



Convertendo uma árvore em regras

- IF *(Outlook = Sunny)* $\wedge$ *(Humidity = High)*
  THEN *PlayTennis = No*

- IF *(Outlook = Sunny)* $\wedge$ *(Humidity = Normal)*
  THEN *PlayTennis = YES*

  .....

Porquê Regras ?

- Permite eliminar um teste numa regra, mas pode reter o teste em outra regra.
- Elimina a distinção entre testes perto da raiz e testes perto das folhas.
- Maior grau de interpretabilidade.

Referências

- Machine Learning. Tom Mitchell. McGraw-Hill.1997.