

Árvores de Decisão

João Gama

Jgama@ncc.up.pt

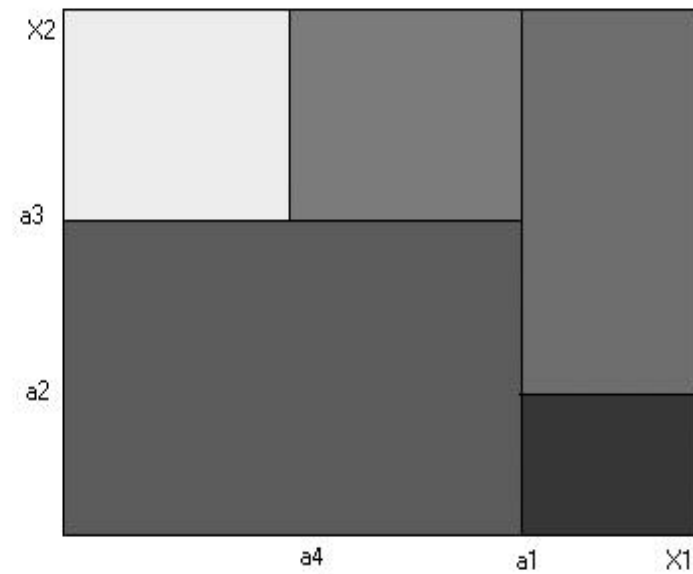
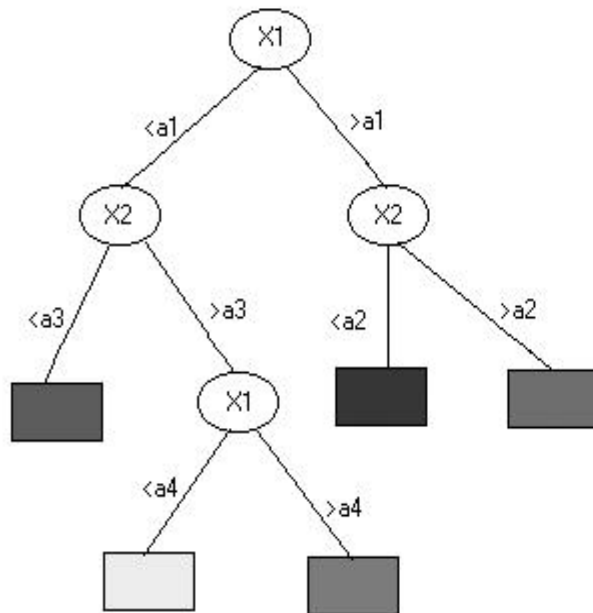
Sumario

- Árvores de decisão
 - Motivação
 - Construção de uma árvore de decisão
 - Critérios para seleccionar atributos
 - Entropia
 - Podar a árvore
 - Estimativas de erro
 - Extensões
 - Árvores multivariadas

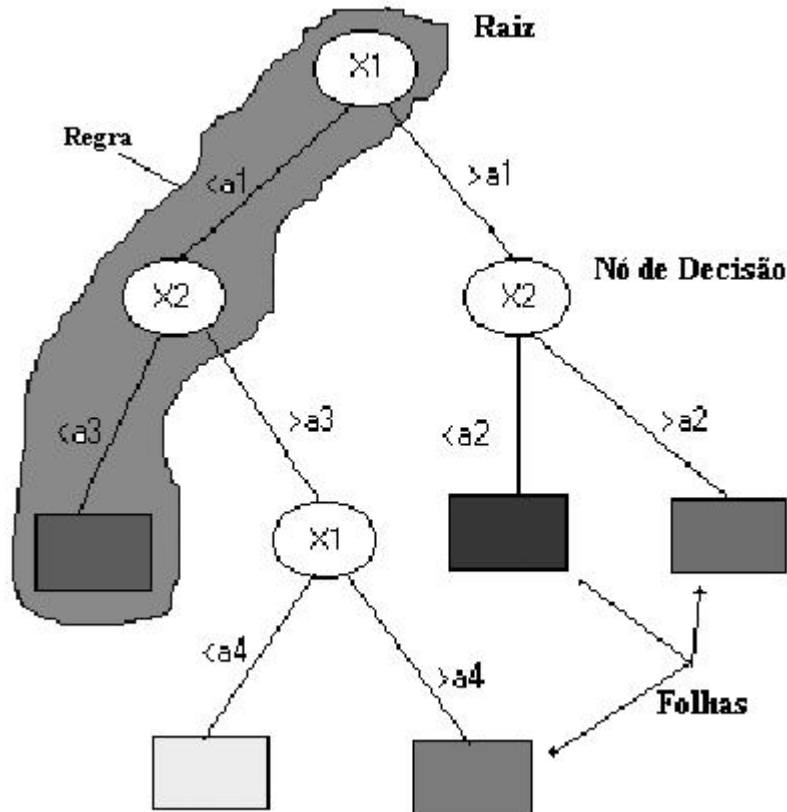
Árvores de Decisão

- Uma árvore de decisão utiliza uma estratégia de *dividir-para-conquistar*:
 - 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.
- Crescente interesse
 - CART (Breiman, Friedman, et.al.)
 - C4.5 (Quinlan)
 - S_{plus}, Statistica, SPSS

Árvores de decisão – Exemplo da partição do espaço dos atributos



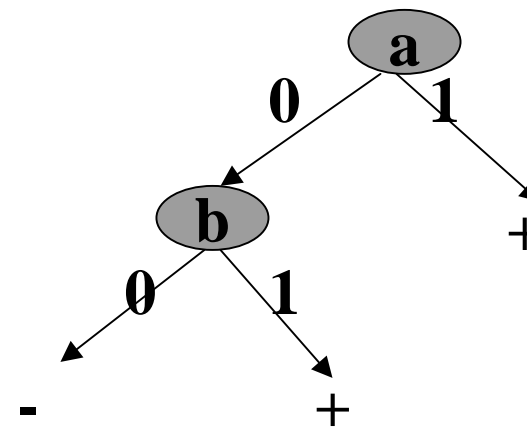
O que é uma Arvore de Decisão?



- Representação por arvores de decisão:
 - Cada nó de decisão contém 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.
- No espaço definido pelos atributos:
 - Cada folha corresponde a uma região
 - Hiper-rectângulo
 - A intersecção dos hiper-rectângulos é vazio
 - A união dos hiper-rectângulos é o espaço completa.

Representação

- Uma árvore de decisão representa a disjunção de conjunções de restrições nos valores dos atributos
 - Cada ramo na árvore é uma conjunção de condições
 - O conjunto de ramos na árvore são disjuntos
 - DNF (disjuntive normal form)
 - Qualquer função lógica pode ser representada por um árvore de decisão.
 - Exemplo **a or b**



Construção de uma árvore de decisão

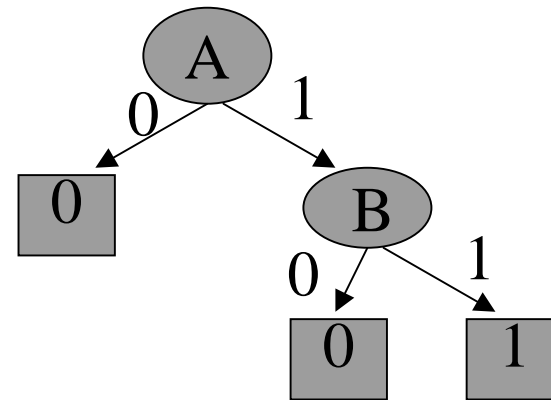
- A ideia *base*:
 1. Escolher um atributo.
 2. Estender a árvore adicionando um ramo para cada valor do atributo.
 3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido)
 4. Para cada folha
 1. Se todos os exemplos são da mesma classe, associar essa classe à folha
 2. Senão repetir os passos 1 a 4

Exemplos de Árvores de Decisão

- Atributos binários

1. And

$A \wedge B$		
0	0	0
0	1	0
1	0	0
1	1	1



1. Exercícios:

1. Representar Or, Xor

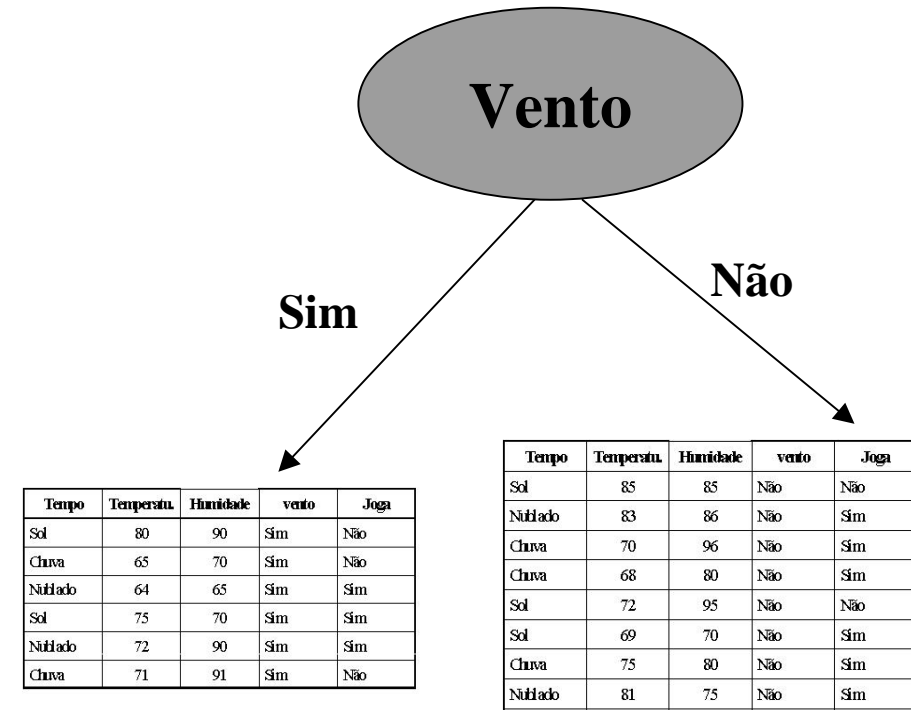
2. $(A \wedge B) \vee (C \wedge D)$

Exemplos:

O conjunto de dados original:

Tempo	Temperatu.	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

Selecciona um atributo:



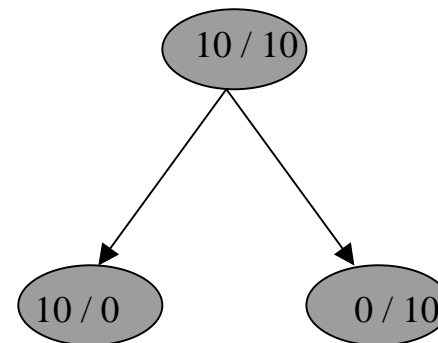
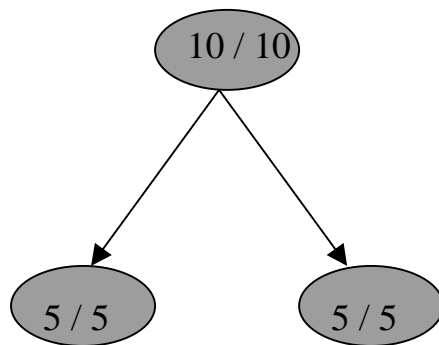
Qual o 'melhor' atributo?

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 mantém 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.



Caracterização das medidas de partição

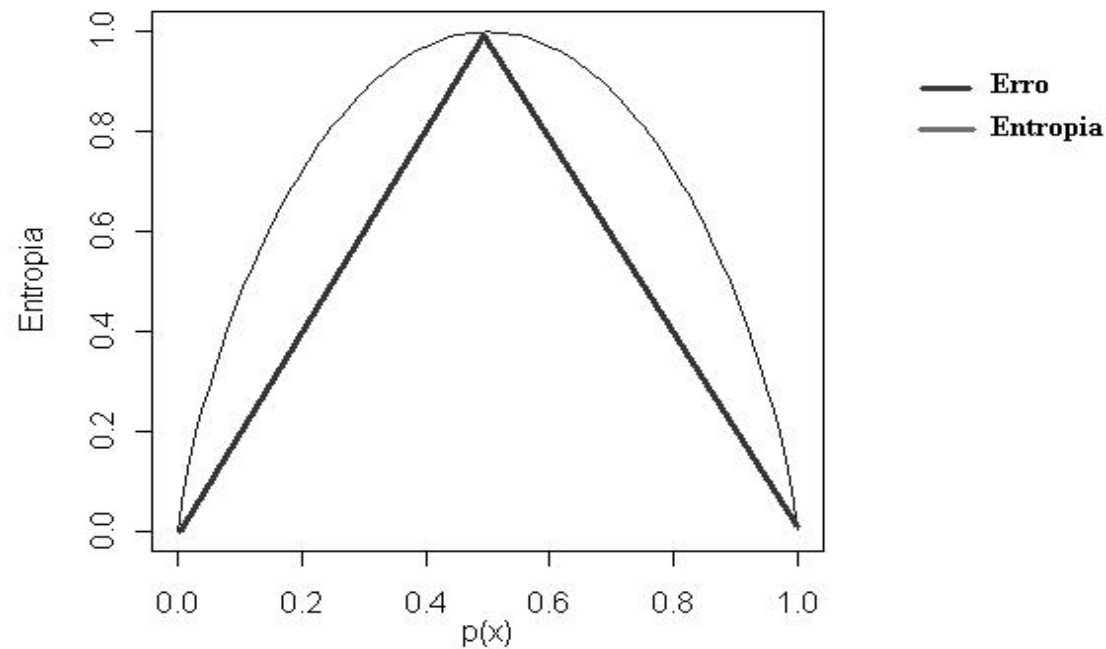
- Medida da diferença dada por uma função baseada nas proporções das classes entre o nó corrente e os nós descendentes.
 - Valoriza a pureza das partições.
 - Gini, entropia
- Medida da diferença dada por uma função baseada nas proporções das classes entre os nós descendentes.
 - Valoriza a disparidade entre as partições.
 - Lopez de Mantaras
- Medida de independência
 - Medida do grau de associação entre os atributos e a classe.

Entropia

- Entropia é uma medida da aleatoriedade de uma variável.
- A entropia de uma variável nominal X que pode tomar i valores:

$$\text{entropia}(X) = -\sum_i p_i * \log_2 p_i$$

- A entropia tem máximo ($\log_2 i$) 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 $0 * \log_2 0 = 0$



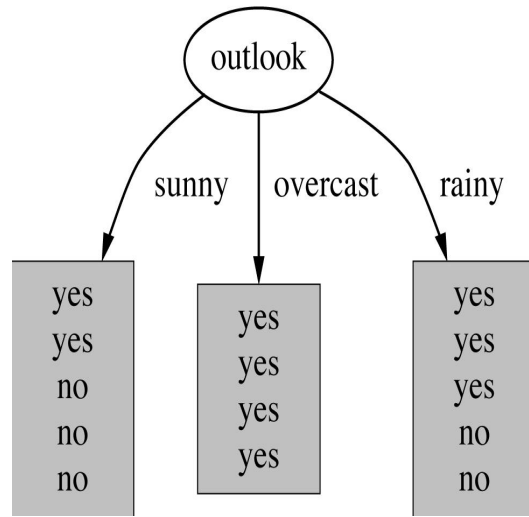
Ganho de Informação

- 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.

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

- A construção de uma árvore de decisão é guiada pelo objectivo de diminuir a entropia ou seja a aleatoriedade -dificuldade de previsão- da variável objectivo.

Calculo do Ganho de Informação de um atributo nominal



	Sol	Nublado	Chuva
Sim	2	4	3
Não	3	0	2

- Informação da Classe:
 - $p(\text{sim}) = 9/14$
 - $p(\text{não}) = 5/14$
 - $\text{Info}(\text{joga}) =$
 - $= -9/14 \log_2 9/14 - 5/14 \log_2 5/14 = 0.940 \text{ bits}$
- Informação nas partições:
 - $p(\text{sim}|\text{tempo}=\text{sol}) = 2/5$
 - $p(\text{não}|\text{tempo}=\text{sol}) = 3/5$
 - $\text{Info}(\text{joga}|\text{tempo}=\text{sol})$
 - $= -2/5 \log_2 2/5 - 3/5 \log_2 3/5 = 0.971 \text{ bits}$
 - $\text{Info}(\text{joga}|\text{tempo}=\text{nublado}) = 0.0 \text{ bits}$
 - $\text{Info}(\text{joga}|\text{tempo}=\text{chuva}) = 0.971 \text{ bits}$
 - **Info(tempo)**
 - $= 5/14 * 0.971 + 4/14 * 0 + 5/14 * 0.971 = 0.693 \text{ bits}$
- Ganho de Informação obtida neste atributo:
 - $\text{Ganho}(\text{tempo}) = 0.940 - 0.693 = 0.247 \text{ bits}$

Calculo do Ganho para Atributos numéricos

- Um teste num atributo numérico produz uma partição binária do conjunto de exemplos:
 - Exemplos onde $\text{valor_do_atributo} < \text{ponto_referência}$
 - Exemplos onde $\text{valor_do_atributo} \geq \text{ponto_referência}$
- Escolha do ponto de referência:
 - Ordenar os exemplos por ordem crescente dos valores do atributo numérico.
 - Qualquer ponto intermédio entre dois valores diferentes e consecutivos dos valores observados no conjunto de treino pode ser utilizado como possível ponto de referência.
 - É usual considerar o valor médio entre dois valores diferentes e consecutivos.
 - Fayyad 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.

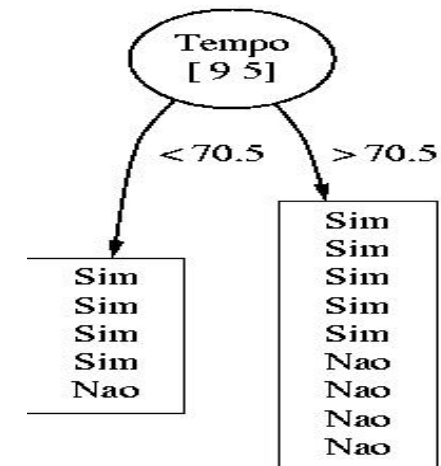
Calculo do Ganho para Atributos numéricos

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

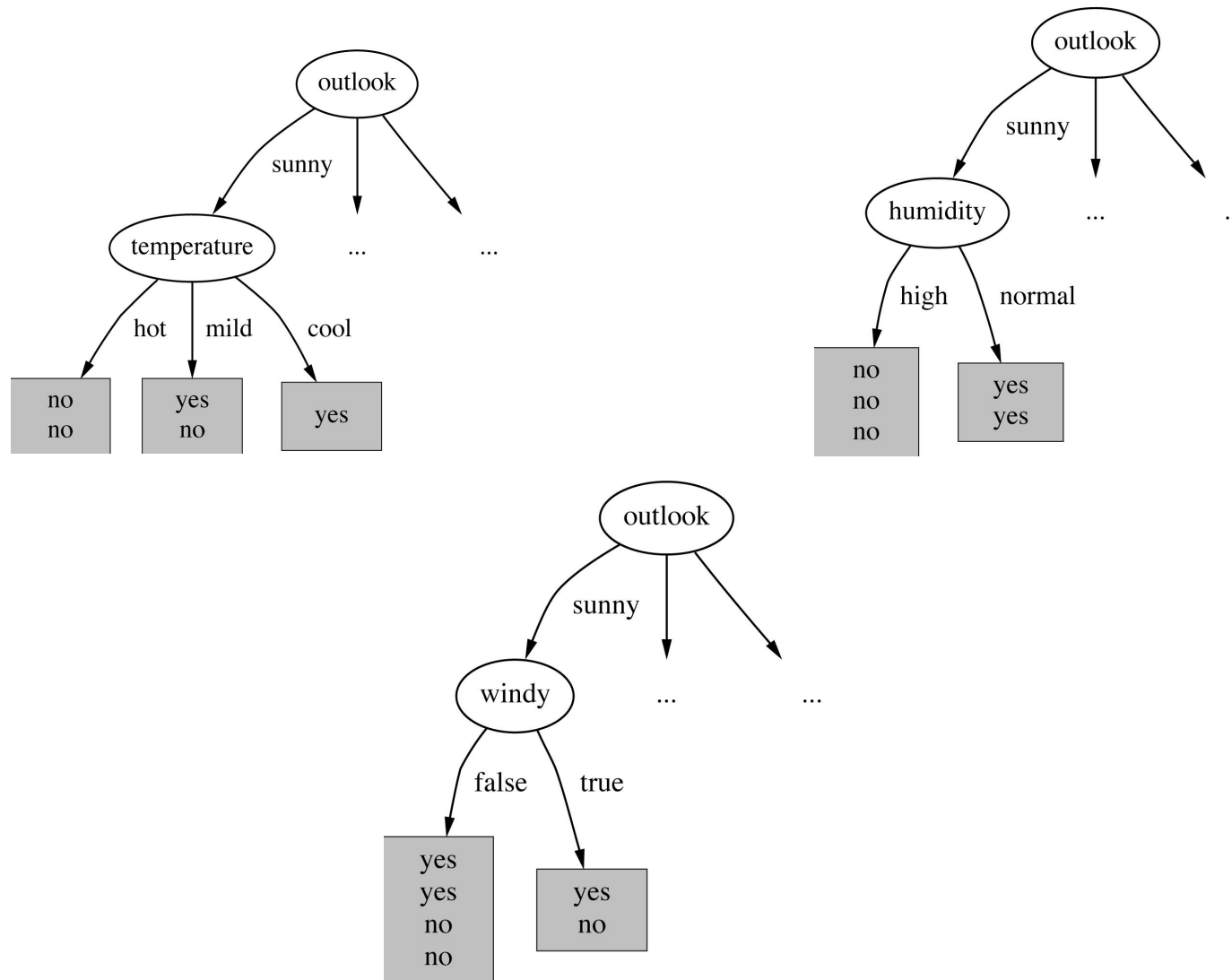
- Considere o ponto de referência temperatura = 70.5
- Um teste usando este ponto de referência divide os exemplos em duas partições:
 - Exemplos onde temperatura < 70.5
 - Exemplos onde temperatura > 70.5
- Como medir o ganho de informação desta partição?

Informação nas partições

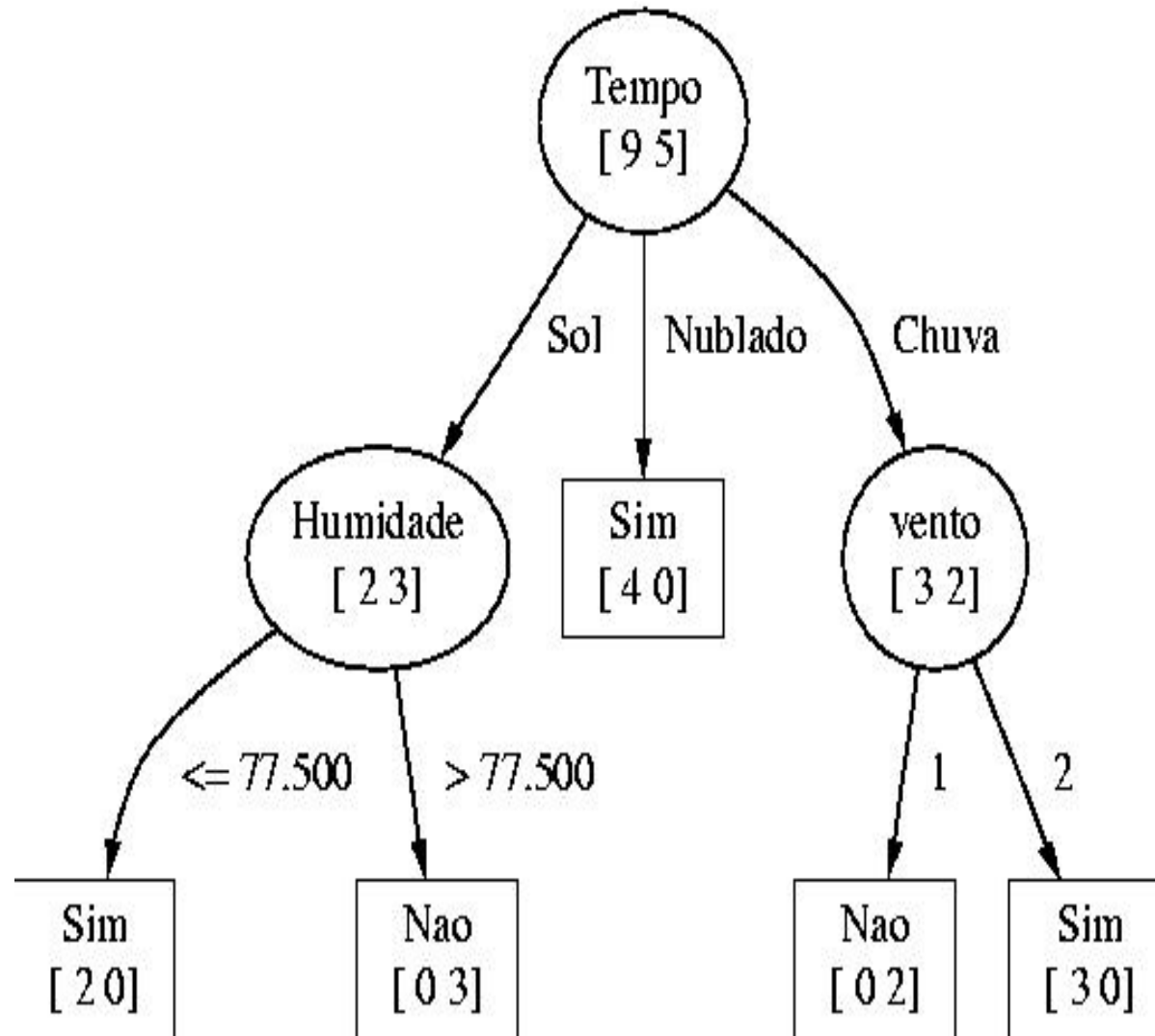
- $p(\text{sim} \mid \text{temperatura} < 70.5) = 4/5$
- $p(\text{não} \mid \text{temperatura} < 70.5) = 1/5$
- $p(\text{sim} \mid \text{temperatura} > 70.5) = 5/9$
- $p(\text{não} \mid \text{temperatura} > 70.5) = 4/9$
- $\text{Info}(\text{joga} \mid \text{temperatura} < 70.5) =$
 - $-4/5 \log_2 4/5 - 1/5 \log_2 1/5 = 0.721$ bits
- $\text{Info}(\text{joga} \mid \text{temperatura} > 70.5) =$
 - $-5/9 \log_2 5/9 - 4/9 \log_2 4/9 = 0.991$ bits
- $\text{Info}(\text{temperatura}) = 5/14 * 0.721 + 9/14 * 0.991 = 0.895$ bits
- $\text{Ganho}(\text{temperatura}) = 0.940 - 0.895 = 0.045$ bits



Repetir o processo



Arvore de decisão final



Critérios de paragem

- Quando parar a divisão dos exemplos?
 - Todos os exemplos pertencem á 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 Arvore de Decisão

- Input: Um conjunto exemplos
- Output: Uma arvore de decisão
- Função GeraArvore(Exs)
 - Se $\text{criterio_paragem(Exs)} = \text{TRUE}$
 - retorna Folha
 - Escolhe o atributo que maximiza o $\text{critério_divisão(Exs)}$
 - Para cada partição i dos exemplos baseada no atributo escolhido
 - $\text{Arvore}_i = \text{GeraArvore(Exs}_i)$
 - Retorna um nó de decisão baseado no atributo escolhido e com descendentes Arvore_i .
- Fim

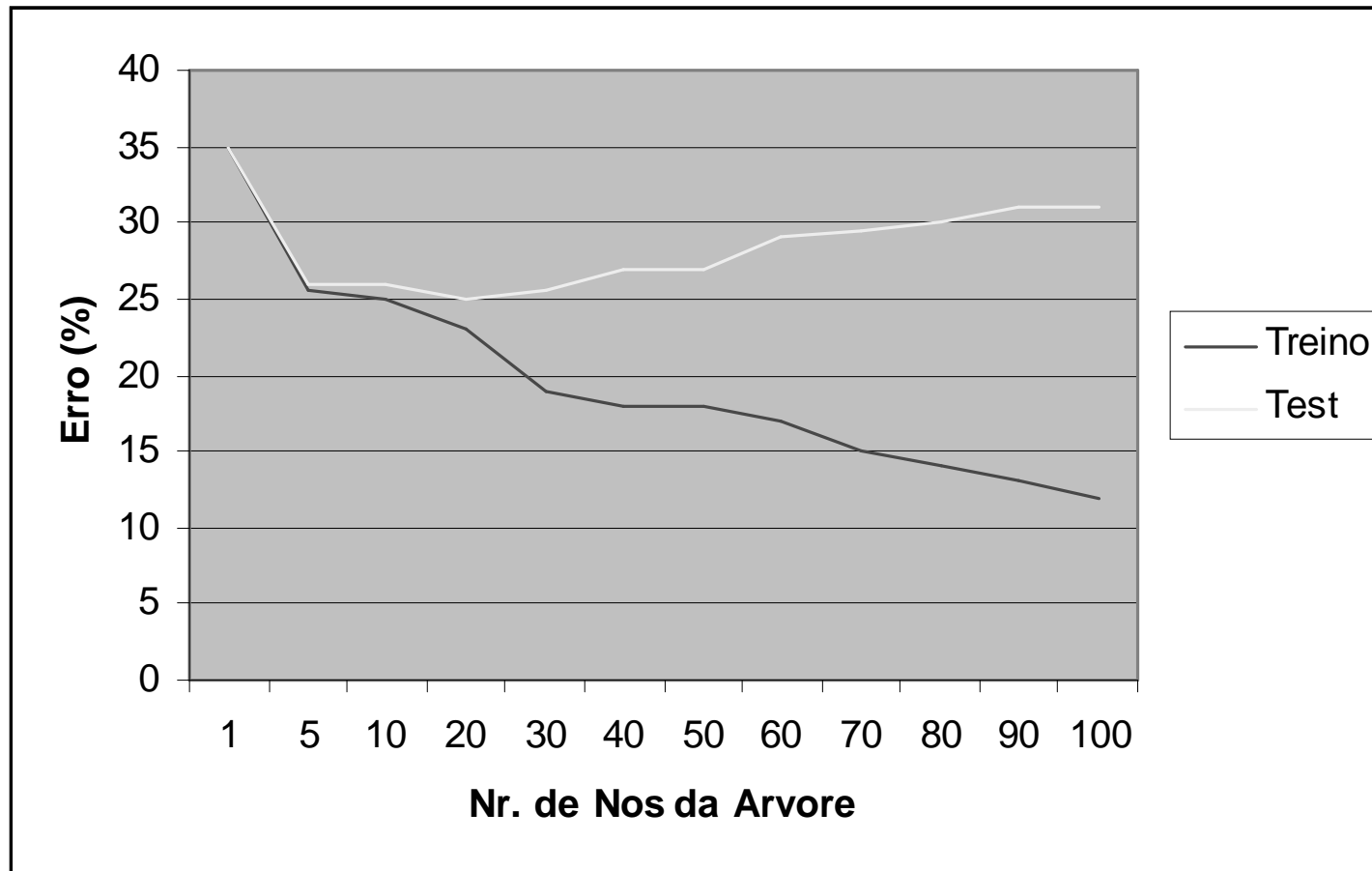
Construção de uma Arvore de Decisão

- O problema de construir uma arvore de decisão:
 - Consistente com um conjunto de exemplos
 - Com o menor numero de nós
 - É um problema *NP* completo.
- Dois problemas:
 - Que atributo seleccionar para teste num nó?
 - Quando parar a divisão dos exemplos ?
- Os algoritmos mais divulgados:
 - Utilizam heurísticas que tomam decisões olhando para a frente um passo.
 - Não reconsideram as opções tomadas.

Sobre-ajustamento

- O algoritmo de partição recursiva do conjunto de dados gera estruturas que podem obter um ajuste aos exemplos de treino perfeito.
 - Em domínios sem ruído o nr. de erros no conjunto de treino 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 nr. de nós



Sobre-ajustamento (“*overfitting*”)

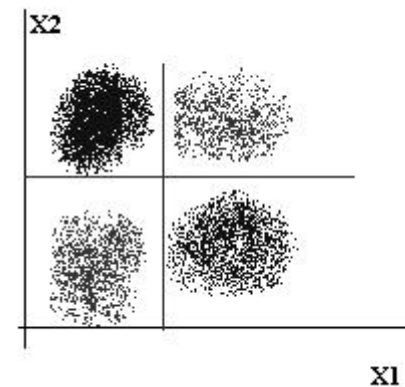
- Definição:
 - Uma árvore de decisão d faz sobre-ajustamento aos dados se existir uma árvore d' tal que:
 - d tem menor erro que d' no conjunto de treino
 - mas d' tem menor erro na população.
- Como pode acontecer:
 - Ruído nos dados
 - Excesso de procura
- O número de parâmetros de uma árvore de decisão cresce linearmente com o número de exemplos.
 - Uma árvore de decisão pode obter um ajuste perfeito aos dados de treino.

Sobre-ajustamento

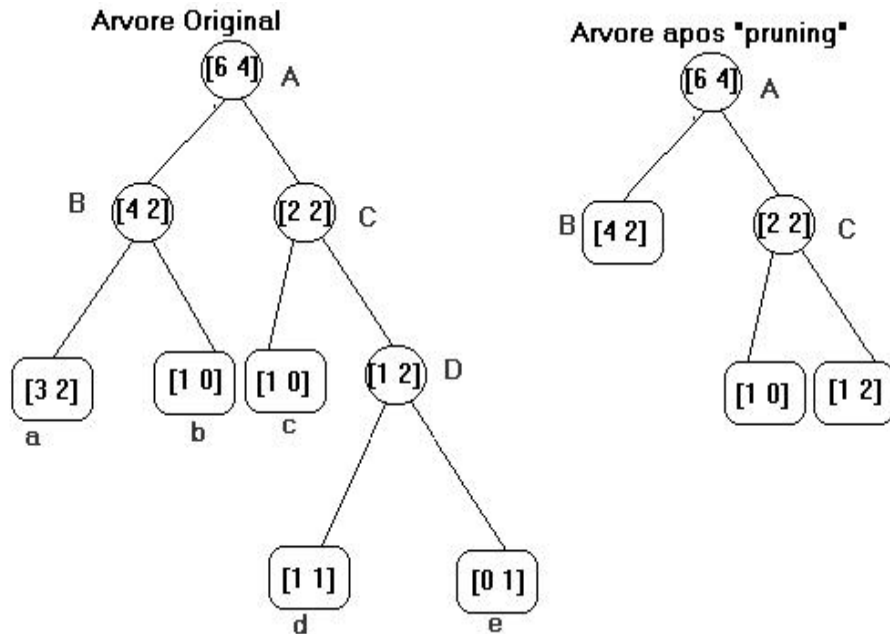
- Occam's razor: preferência pela hipótese mais simples.
 - Existem menos hipóteses simples do que complexas.
 - Se uma hipótese simples explica os dados é pouco provável que seja uma coincidência.
 - Uma hipótese complexa pode explicar os dados apenas por coincidência.
- A avaliação de uma hipótese deve ter em conta o processo de construção da hipótese.

Simplificar a árvore

- Duas possibilidades:
 - Parar o crescimento da árvore mais cedo (pre-pruning).
 - Crescer uma árvore completa e podar a árvore (pos-pruning).
 - “*Growing and pruning is slower but more reliable*”
 - Quinlan, 1988
 - O problema do “Xor”
 - Requer olhar em frente mais que um nível.



Um algoritmo básico de pruning



- Percorre a arvore 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.
- 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

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

Estimativas de Erro

- O problema fundamental do algoritmo de poda é a estimativa de erro num determinado nó.
 - O erro estimado a partir do conjunto de treino não é um estimador fiável.
- O “*reduced error pruning*”
 - consiste em obter estimativas de erro a partir de um conjunto de validação independente do conjunto de treino.
 - Reduz o volume de informação disponível para crescer a árvore.
- O “*Cost complexity pruning*”, Breiman, 1984
 - Podar com base na estimativa do erro e complexidade da árvore.
 - Cart (Breiman et al.)
- O “*Error based pruning*”,
 - Podar com base numa estimativa do erro no conjunto de treino.
 - Assume uma distribuição Binomial para os exemplos de um nó.
 - Usado no C5.0

O “Error based pruning”

- Um determinado nó contém N exemplos.
 - Classificando estes exemplos utilizando a classe majoritária, vão ser mal classificados E exemplos.
 - O erro neste nó é E/N .
- Qual é o verdadeiro erro p neste nó ?
- Assumindo que:
 - Os exemplos de um nó constituem uma amostra de uma população que segue uma distribuição binomial.
 - A variância do erro neste nó é dada por:
 - $P * (1 - P) / N$
 - P é o erro na população (desconhecido).
- Fixando um nível de confiança, podemos obter um intervalo de confiança $[U_{\min}:U_{\max}]$ para p .
- U_{\max} é uma estimativa pessimista para o erro neste nó.

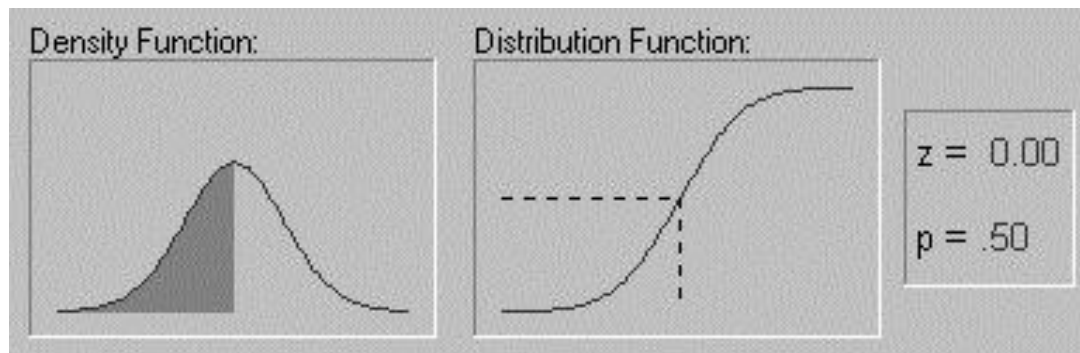
O “Error based pruning”

- A probabilidade de uma variável aleatória X , de média 0 e desvio padrão 1, assumir um valor maior que z , com confiança c , é:
 - $P[X \geq z] = c$
 - Para um determinado nível de confiança c , é determinado o valor superior para o intervalo de confiança

$$p\left[\frac{f - p}{\sqrt{p(1-p)/N}} \geq z\right] = c \quad \text{onde } f = \frac{\text{Erros}}{N}$$

$$p = \left(f + \frac{z^2}{2N} + z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}\right) / \left(1 + \frac{z^2}{N}\right)$$

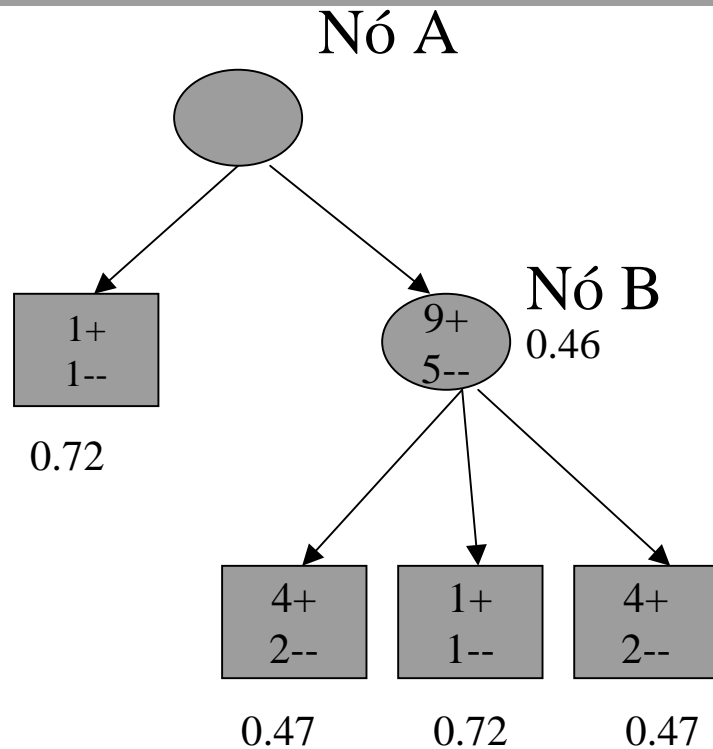
Pr($X > z$)	z
0,1%	3,09
0,5%	2,58
1,0%	2,33
5,0%	1,65
10%	1,28
20%	0,84
25%	0,69
40%	0,25



Um algoritmo de poda

- Percorre a árvore em profundidade
- Para cada nó de decisão calcula:
 - Uma estimativa *pessimista* do erro no nó
 - Soma das estimativas pessimistas dos erros nos nós descendentes
- Se a estimativa do erro no nó é menor ou igual à soma das estimativas de erros dos nós descendentes o nó é transformado em folha.

Exemplo – “Error based pruning”



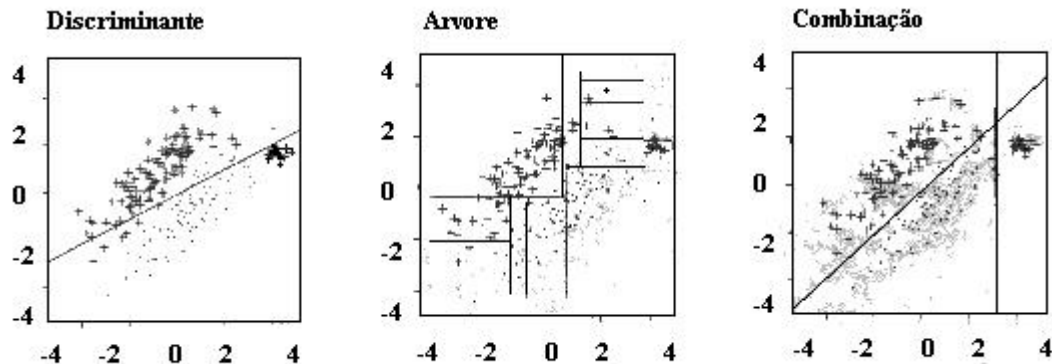
- No nó B
 - Estimativa no nó: 0.46
 - Soma pesada das estimativas nas folhas:
 - $6/14*0.47+2/14*0.72+6/14*0.47=0.51$
 - Como $0.46 < 0.51$ o nó é transformado numa folha.
- Exercício:
 - Qual a decisão para o nó A ?

Valores Desconhecidos

- Pré-Processados
 - Substituir o valor desconhecido pelo valor mais provável
 - Atributos numéricos: média.
 - Atributos nominais: mediana.
- Na construção do modelo
 - Assumir que um atributo tem como possível valor o valor desconhecido.
 - Atribuir um peso a cada exemplo.
 - Nos exemplos em que o atributo de teste toma um valor desconhecido, o exemplo é passado para todos os nós descendentes com um peso proporcional á probabilidade de um exemplo seguir o ramo.

Arvores Multivariadas

- Motivação
 - Discriminante linear
 - Superfícies de decisão obliquas em relação aos eixos definidos pelos atributos.
 - Arvores de Decisão
 - Partição do espaço dos atributos.
 - Superfícies de decisão: hiper-rectangulos.
 - Arvores multivariadas:
 - Combinação de superficies de decisão obliquas com partição do espaço dos atributos:



Árvores Multivariadas

- Discriminantes Recursivos
 - QUEST
 - Em cada nó constrói um discriminante linear.
- Árvores Multivariadas
 - LMDT
 - Em cada nó constrói uma “Maquina Linear” que é usada como teste neste nó.
 - LTREE
 - Em cada nó constrói um discriminante linear
 - Todos os exemplos neste nó são estendidos com novos atributos
 - Cada novo atributo é a probabilidade de o exemplo estar num lado do hiperplano.
 - A capacidade de discriminação de um novo atributo é estimada em competição com os atributos originais.

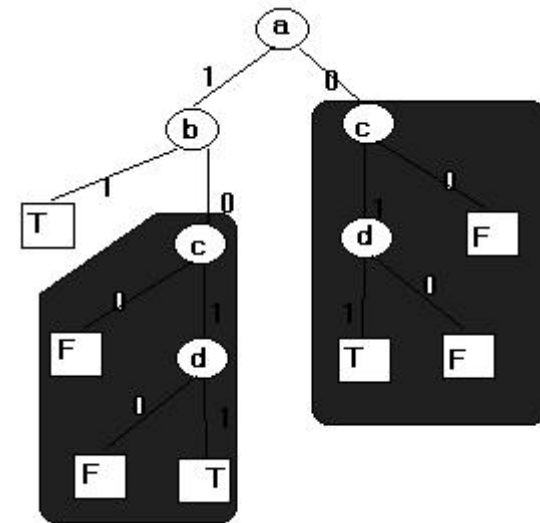
Vantagens das Árvores de decisão

- Método não-paramétrico
 - Não assume nenhuma distribuição particular para os dados.
 - Pode construir modelos para qualquer função desde que o número de exemplos de treino seja suficiente.
- A estrutura da árvore de decisão é independente da escala das variáveis.
 - Transformações monótonas das variáveis ($\log x$, $2*x$, ...) não alteram a estrutura da árvore.
- Elevado grau de interpretabilidade
 - Uma decisão complexa (prever o valor da classe) é decomposto numa sucessão de decisões elementares.
- É eficiente na construção de modelos:
 - Complexidade média $O(n \log n)$
- Robusto á presença de pontos extremos e atributos redundantes ou irrelevantes.
 - Mecanismo de seleção de atributos.

Inconvenientes das Árvores de decisão

- Instabilidade
 - Pequenas perturbações do conjunto de treino podem provocar grandes alterações no modelo aprendido.
- Presença de valores desconhecidos
- Fragmentação de conceitos
 - Replicação de sub-árvores

$$(a \wedge b) \vee (c \wedge d)$$



Bibliografia Adicional

- Online:
 - <http://www.Recursive-Partitioning.com/>
- Tom Mitchell
 - Machine Learning (chap.3)
 - MacGrawHill, 1997
- Quinlan, R.
 - “C4.5 Programs for Machine Learning”
 - Morgan Kaufmann Publishers, 1993
- L.Breiman, J.Friedman, R.Olshen, C.Stone
 - “Classification and Regression Trees”
 - Wadsworth, 1984

Exercícios

- Considere um problema de duas classes, definido por 4 atributos binários x_1, x_2, x_3, x_4 .
 - Represente sob a forma de uma árvore o conceito
 - $(x_1 \text{ e } x_2 \text{ e } x_3) \text{ ou } x_4$
 - $(x_1 \text{ e } x_2) \text{ ou } (x_3 \text{ e } x_4)$
- Utilizando ferramentas computacionais, calcule o atributo para a raiz de uma árvore de decisão nos problemas
 - *Iris*
 - *Balance-scale*
- Utilizando ferramentas computacionais, indique uma árvore qualquer (a escolha do atributo a usar em cada nó pode ser aleatória) para o problema *Iris*
 - Qual a taxa de erro da árvore no *dataset* completo?
 - Aplique o algoritmo de poda á árvore.