Aprendizado de Máquina

Aula 4: Algoritmos Baseados em Procura

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br

Tópicos

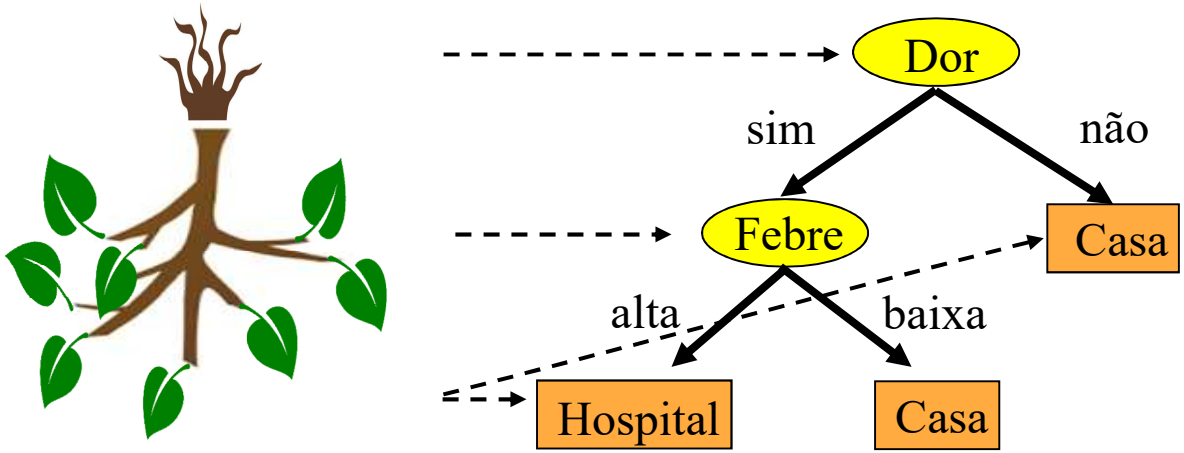
- Árvores de decisão
- Algoritmo de Hunt
- Medidas para escolha de atributos
- Ponto de referência
- Critério de parada
- Espaço de hipóteses

Introdução

- Explicação das decisões pode ser importante para algumas aplicações
  - Redes Neurais e Máquinas de Vetores de Suporte são caixas pretas
- Modelos interpretáveis são gerados por algumas algoritmos de AM
  - Algoritmos que induzem
    - Árvores de características (decisão)
    - Conjunto de regras
  - Naive Bayes

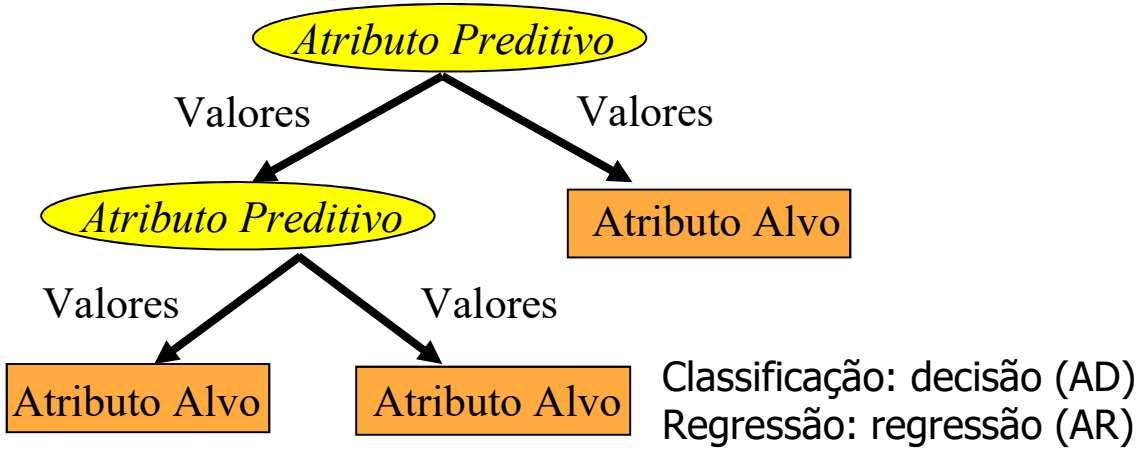
Algoritmos de indução de árvores

- Geram modelos com formato de árvores de características



Árvores de características

- Particionam características (atributos) de forma hierárquica



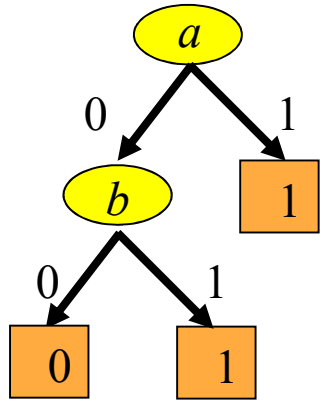
Outro exemplo simples

a	b	a v b
0	0	0
0	1	1
1	0	1
1	1	1

Nós internos e raiz: atributos preditivos
Nós externos (folhas): atributo alvo

Outro exemplo simples

a	b	a v b
0	0	0
0	1	1
1	0	1
1	1	1



Nós internos e raiz: atributos preditivos
Nós externos (folhas): atributo alvo

Algoritmo de indução de AD

- Existem vários, dentre eles:
  - Algoritmo de Hunt
    - Um dos primeiros
    - Base de vários algoritmos atuais
  - CART
  - ID3
  - C4.5
  - VFDT

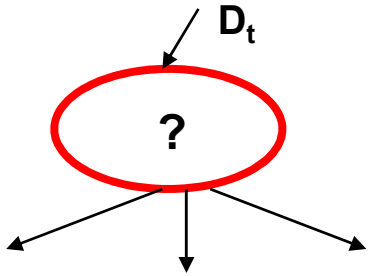
Algoritmo de Hunt

- Seja X_t o conjunto de objetos de treinamento que atingem o nó t

Se todos os objetos de X_t $\in$ a mesma classe y
Então O nó t é um nó folha rotulado pela classe y
Senão Selecionar um atributo preditivo teste para dividir X_t
Dividir X_t em subconjuntos usando valores desse atributo
Aplicar algoritmo a cada subconjunto gerado

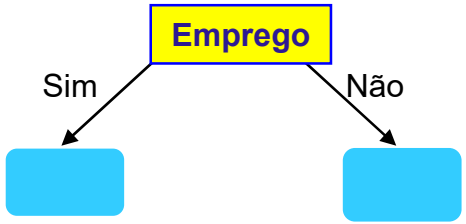
Algoritmo de Hunt

Emprego	Estado	Renda	Inadimplente
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



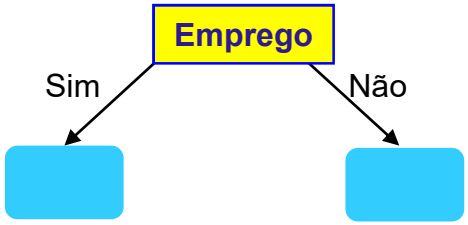
Algoritmo de Hunt

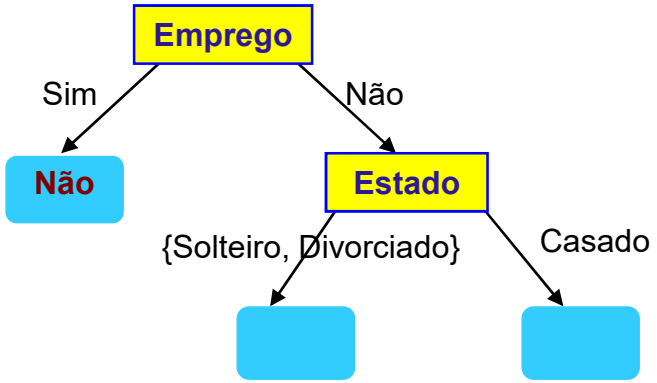
Emprego	Estado	Renda	Inadimplente
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



Algoritmo de Hunt

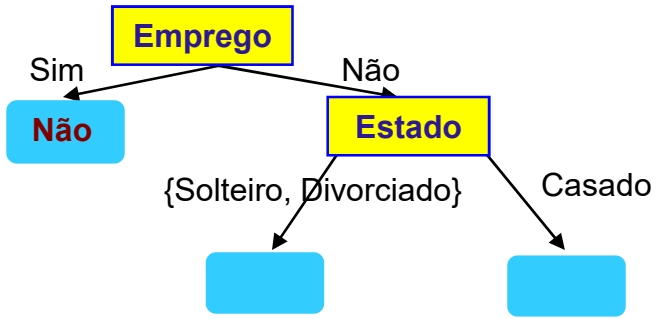
Emprego	Estado	Renda	Inadimplente
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

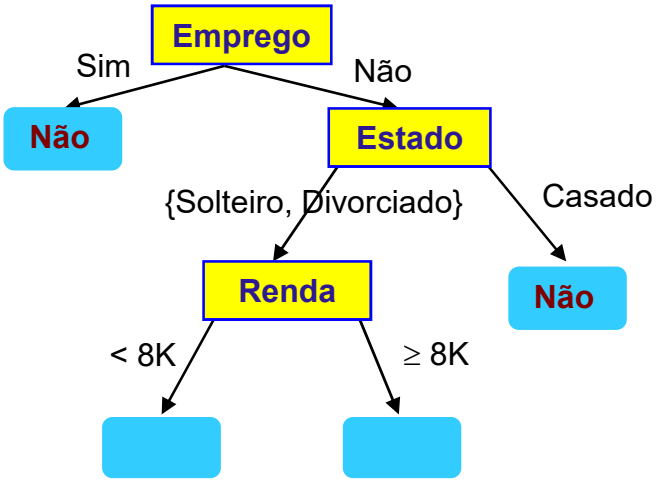




Algoritmo de Hunt

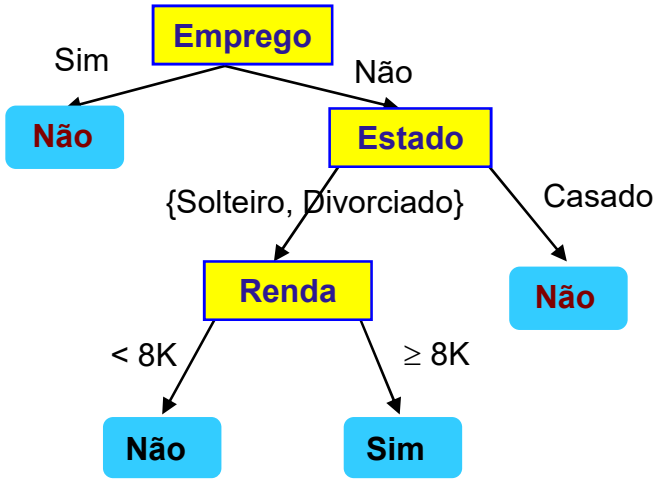
Emprego	Estado	Renda	Inadimplente
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim





Algoritmo de Hunt

Emprego	Estado	Renda	Inadimplente
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



Indução de ADs

- Geralmente usa estratégia gulosa de divisão e conquista
  - Divide progressivamente objetos, cada vez baseado em um atributo preditivo
    - Atributo é escolhido para otimizar algum critério
- Decisões importantes
  - Escolha do atributo preditivo
  - Como dividir os objetos entre os ramos usando o atributo preditivo
  - Quando parar de dividir os objetos

Escolha do atributo preditivo

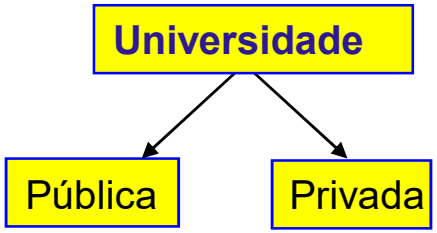
- Atributo preditivo que melhor particiona conjunto atual de objetos
  - Para um mesmo atributo preditivo, diferentes partições podem ser geradas
  - Necessário escolher:
    - Atributo preditivo mais discriminativo
    - Melhor partição para esse atributo
      - Como dividir os objetos que chegam a este nó

Como dividir os objetos

- Depende do tipo do atributo preditivo e do número de divisões a serem geradas
  - Atributo preditivo assume valores binários (atributo binário)
    - Divisão binária
      - Árvore binária
  - Atributo preditivo assume valores n-ários (atributo n-ário)
    - Divisão binária
    - Divisão n-ária (n > 2)

Atributo binário

- Teste mais simples que existe
  - Tem dois possíveis resultados (filhos)



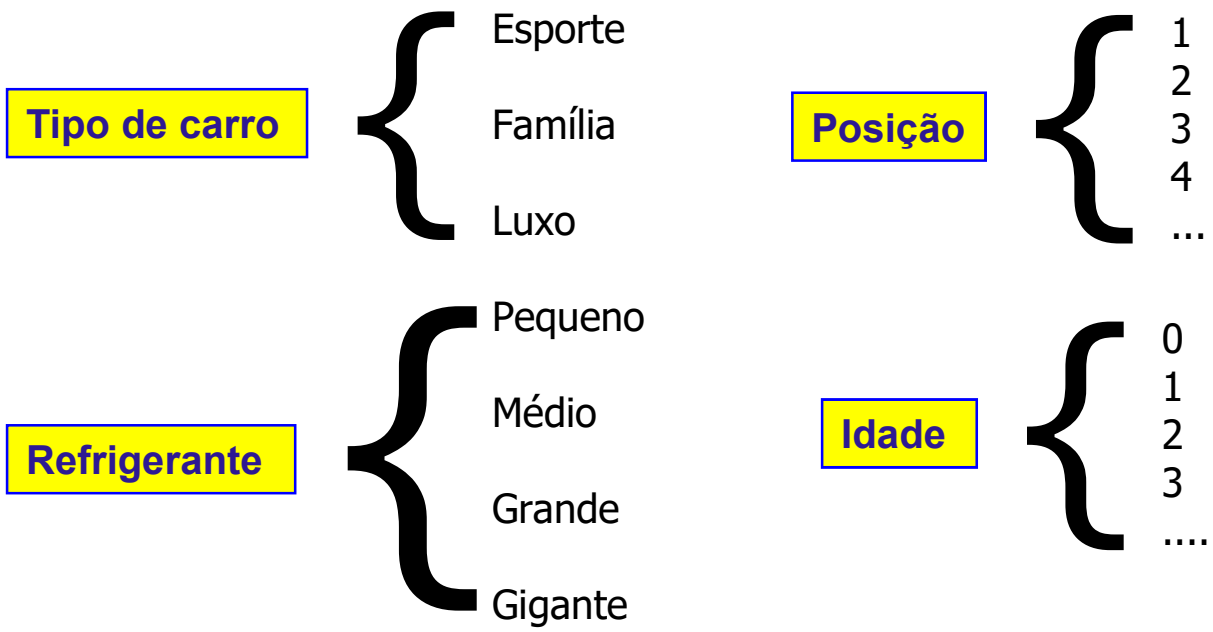
Atributo n-ário

- Divisão
  - Binária
  - N-ária
- Depende do tipo do atributo
  - Simbólico
    - Nominal
    - Ordinal
  - Numérico
    - Discreto
    - Contínuo

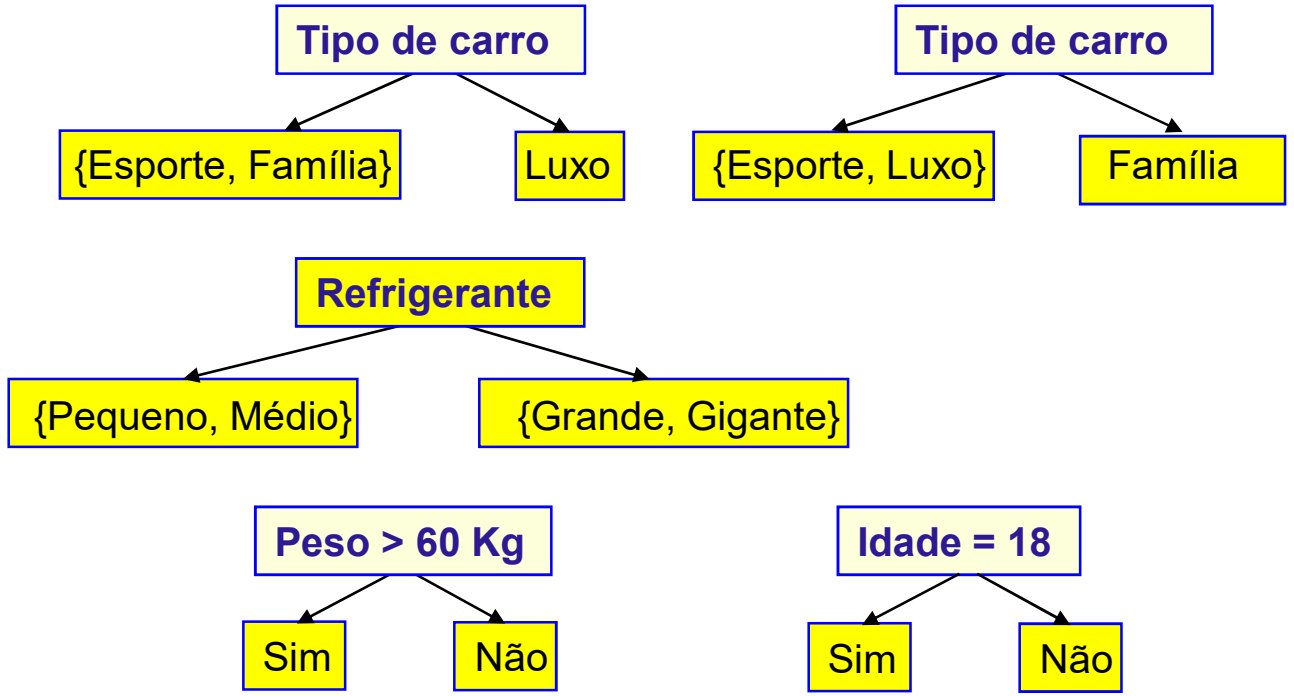
Divisão binária para atributo n-ário

- Único teste com 2 possíveis resultados (filhos)
- Condição de teste é uma comparação
  - Ex.: A < valor, A = valor, A ∈ {valores}, ...
  - Escolher valor(es) que gera(m) melhor partição
    - Ponto de referência
  - Tipo simbólico: grupo de valores em cada ramo
    - Ordinais: valores agrupados não devem violar relação de ordem
    - Nominais: grupos devem fazer sentido
  - Tipo numérico: divide valores em intervalos

Atributos simbólicos X numéricos



Divisão binária para atributo n-ário



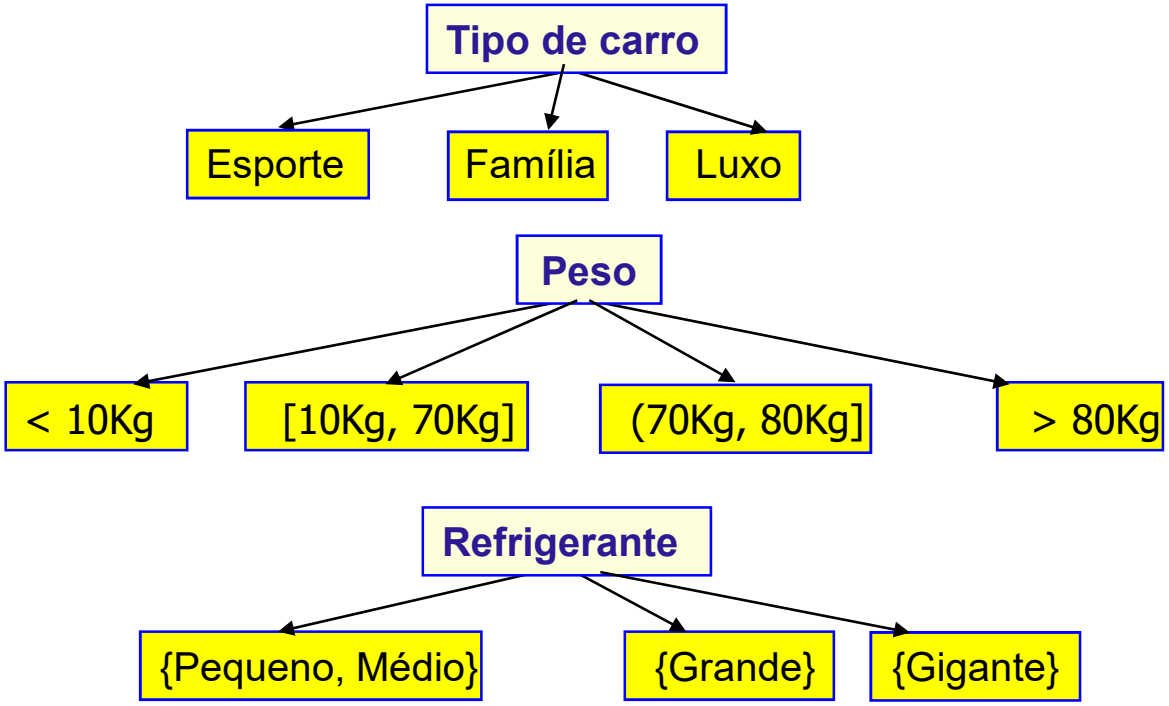
Divisão n-ária para atributo n-ário

- Atributos simbólicos
  - Duas alternativas para definir número de resultados do teste
    - Fazer #ramos = #possíveis valores
    - Agrupar parte dos valores em cada ramo
      - Ordinais
      - Nominais
- Atributos numéricos
  - Dividir valores em N intervalos

Divisão n-ária para atributo n-ário

- Atributos numéricos
  - Condição de teste formada pode ser formado por 1 ou mais comparações
    - Um operador
      - Ex. A < valor, A = valor
    - Mais de um operador
      - Ex.: $valor_{inf} < A < valor_{sup}$
    - Escolher valores (pontos de referência)

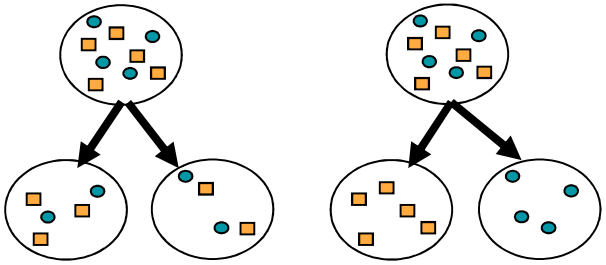
Divisão n-ária para atributo n-ário



Pausa

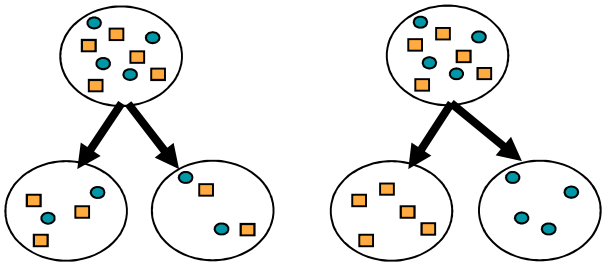
Escolha de atributos preditivos

- Selecionam atributo que melhor discrimina os objetos atuais
  - Buscam partições mais puras após divisão
    - Quanto mais homogêneas as partições, mais puras
  - Medidas de impureza



Medidas de impureza

- Baseadas no grau de impureza dos nós filhos
  - Quando maior a impureza, pior
- Diferentes medidas geram diferentes partições
- Exemplos
  - Entropia
  - Gini
  - Erro de classificação
  - Qui-quadrado



Medidas de impureza

$$Entropia(v) = -\sum_{i=1}^{C} p(i/v)\log_2\ p(i/v)$$

$$Gini(v) = 1 - \sum_{i=1}^{C}\left[p(i/v)\right]^2$$

$$ErroClass\ (v) = 1 - \max_i\left[p(i/v)\right]$$

Onde:

P(i/v) = fração de dados pertencente a classe i em um nó v

C = número de classes

Considera-se que $0\log_2 0 = 0$

Exemplo

- Calcular a medida de impureza Gini para os dados abaixo:

$$Gini(v) = 1 - \sum_{i=1}^{C} [p(i/v)]^2$$

C1	**0**
C2	**6**
Gini=?	

Atributo a

C1	**1**
C2	**5**
Gini=?	

Atributo b

C1	**2**
C2	**4**
Gini=?	

Atributo c

C1	**3**
C2	**3**
Gini=?	

Atributo d

Exemplo

$$Gini(v) = 1 - \sum_{i=1}^{C} [p(i/v)]^2$$

P(C1) = 0/6 = 0 *P(C2) = 6/6 = 1*

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

P(C1) = 1/6 *P(C2) = 5/6*

Gini = 1 – (1/6)² – (5/6)² = 0.278

P(C1) = 2/6 *P(C2) = 4/6*

Gini = 1 – (2/6)² – (4/6)² = 0.444

P(C1) = 3/6 *P(C2) = 3/6*

Gini = 1 – (3/6)² – (3/6)² = 0.500

C1	**0**
C2	**6**
Gini=0.000	

Atributo a

C1	**1**
C2	**5**
Gini=0.278	

Atributo b

C1	**2**
C2	**4**
Gini=0.444	

Atributo c

C1	**3**
C2	**3**
Gini=0.500	

Atributo d

Medida Gini média ponderada

- Usada pelos algoritmos CART, SLIQ, SPRINT
- Quando um nó pai possui *k* filhos, a impureza da divisão é definida por:

$$Gini_{divisão} = \sum_{f=1}^{k} \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

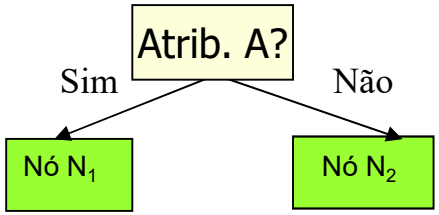
Média ponderada

Onde:

$N(v_f)$: número de objetos no filho (v_f)
$N(v_p)$: número de objetos no pai (v_p)

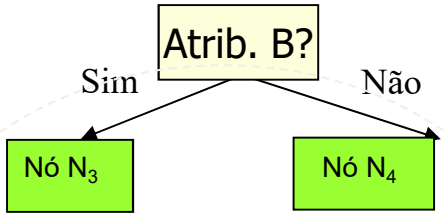
Divisão de atributos binários

	Pai
C1	6
C2	6
Gini = 0.500	



	Nó 1	*Nó 2*
C1	4	2
C2	3	3
Gini =		

$Gini_{divisão}$ =

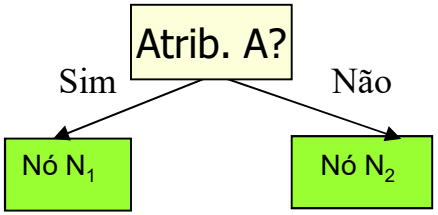


	Nó 3	*Nó 4*
C1	1	5
C2	4	2
Gini =		

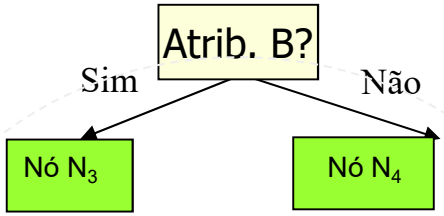
$Gini_{divisão}$ =

Divisão de atributos binários

	Pai
C1	6
C2	6
Gini = 0.500	



	Nó 1	*Nó 2*
C1	4	2
C2	3	3
Gini = 0.486		



	Nó 3	*Nó 4*
C1	1	5
C2	4	2
Gini = 0.375		

$Gini_{divisão}$ = (7/12)x0.49 + (5/12)x0.48
= 0.486

$Gini_{divisão}$ = tarefa de casa
= 0.375

Atributos n-ários

- Várias possíveis posições de referência
- Cada posição tem uma matriz de contagens associada a ela
    - Contagens das proporções das classes em cada uma das partições

Critério de parada

- Diversas alternativas:
  - Os objetos do nó atual têm a mesma classe
  - Os objetos do nó atual têm valores iguais para os atributos de entrada, mas classes diferentes
  - O número de objetos do nó é menor que um dada quantidade
  - Todos os atributos preditivos já foram incluídos no caminho atual

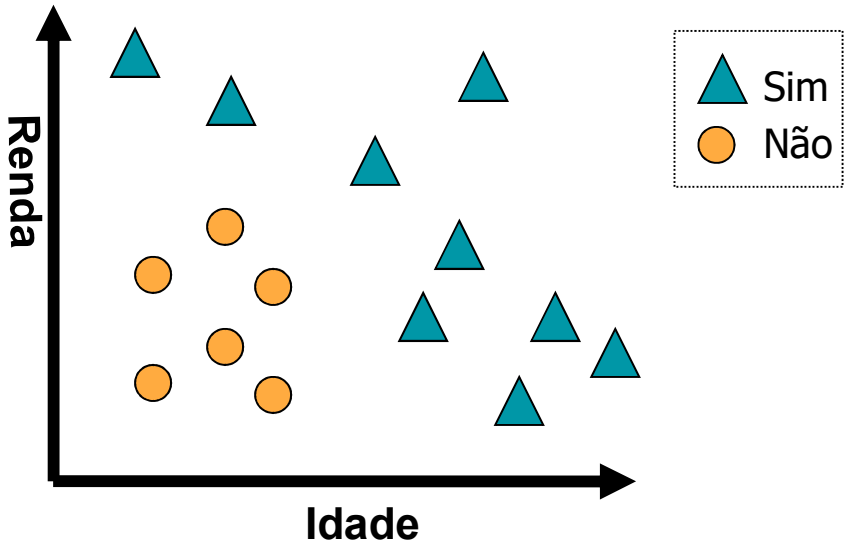
Exemplo

- Sejam os dados abaixo referentes a solicitações de crédito bancário
  - Construir uma árvore de decisão que classifica aplicação para cartão de crédito

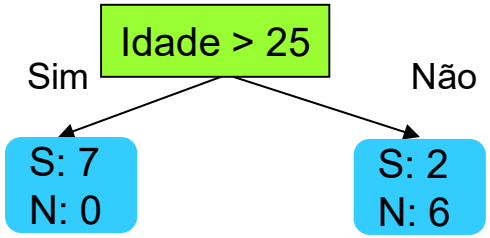
Idade	Renda	Classe
20	2000	Sim
30	5200	Não
60	5000	Sim
40	6000	Não
...		

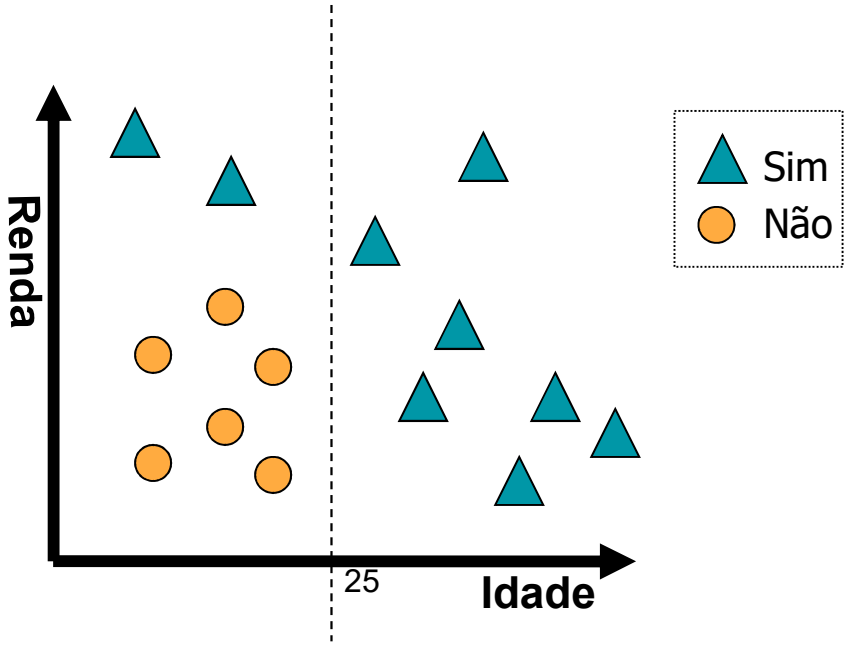
Busca no espaço de hipóteses

- Construir uma AD que classifica solicitante de cartão de crédito
  - Aprova (Sim)
  - Não aprova (Não)
- Atributos preditivos
  - Idade
  - Renda

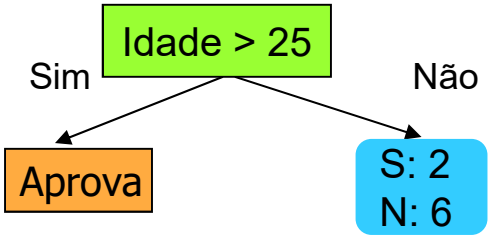


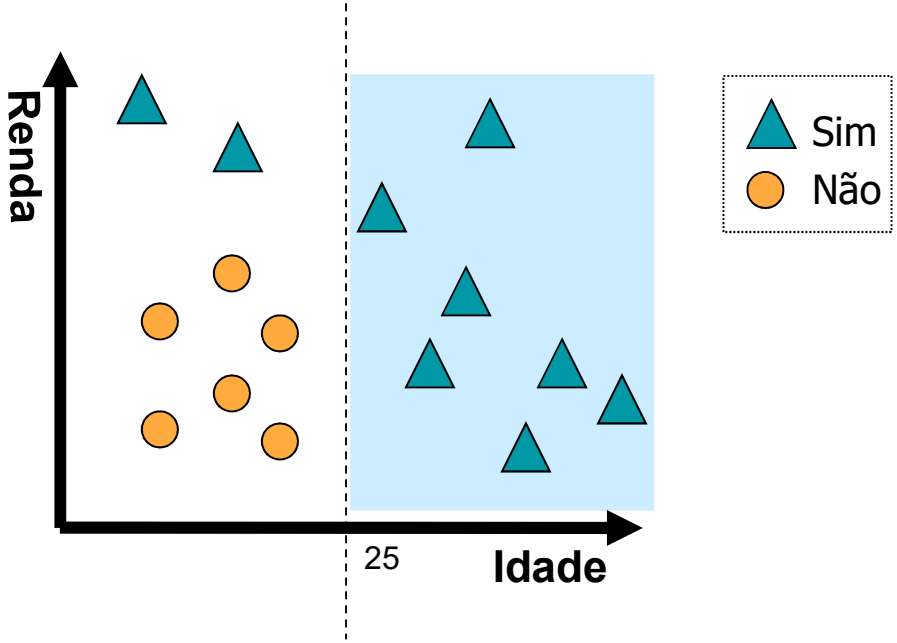
Busca no espaço de hipóteses



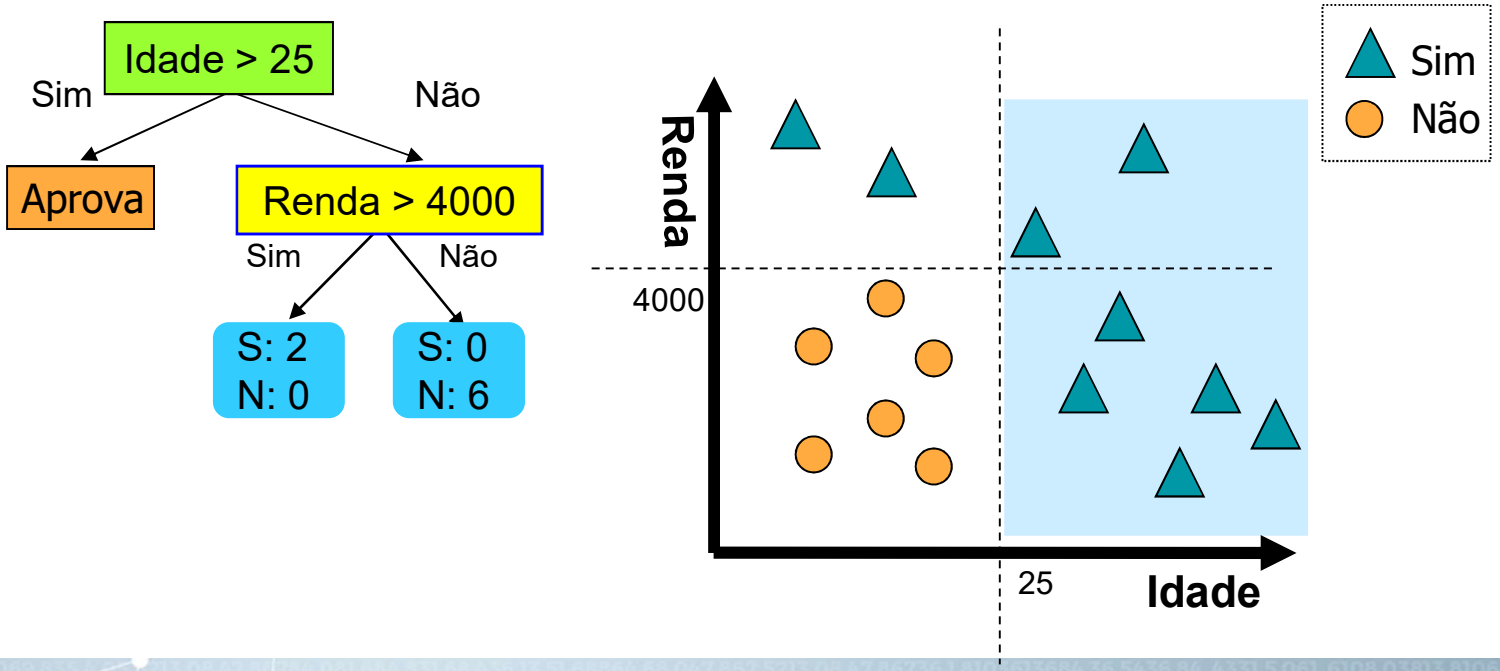


Busca no espaço de hipóteses

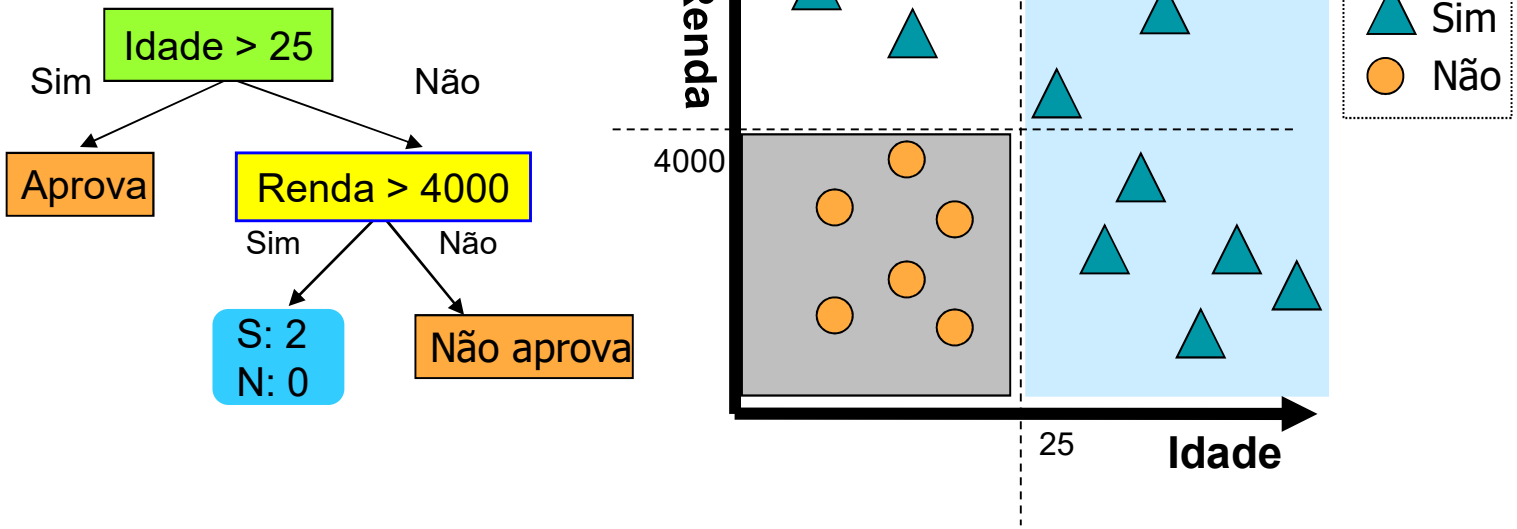




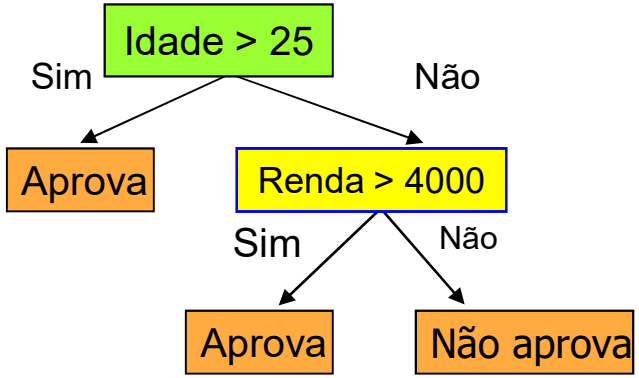
Busca no espaço de hipóteses

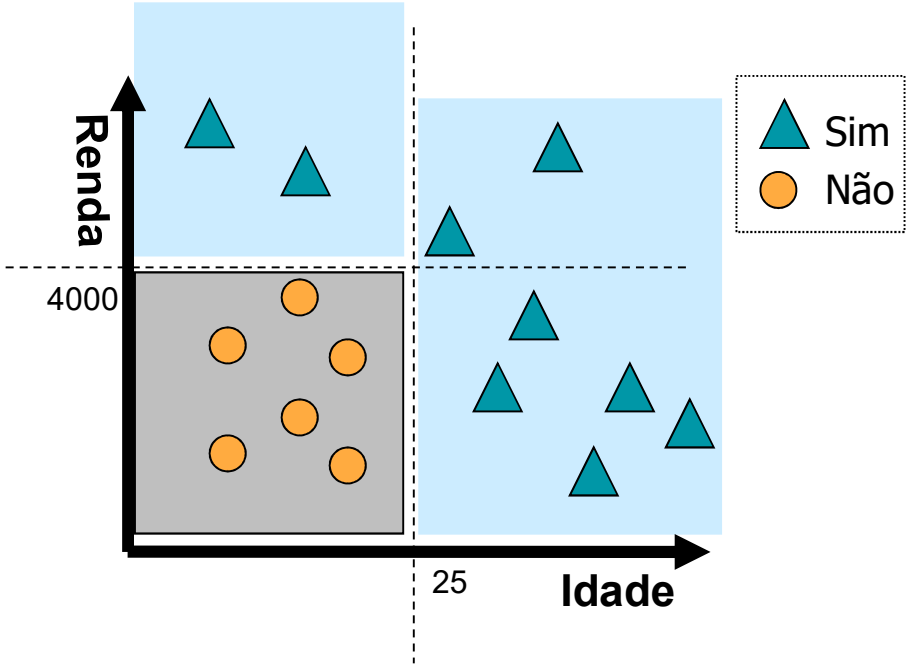


Busca no espaço de hipóteses



Busca no espaço de hipóteses

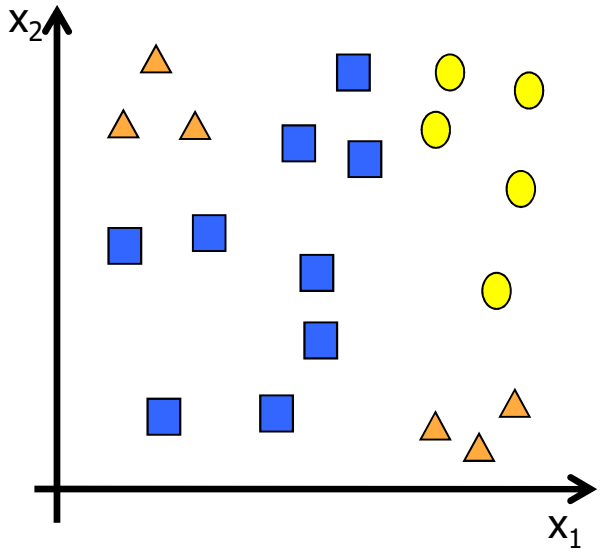




Pausa

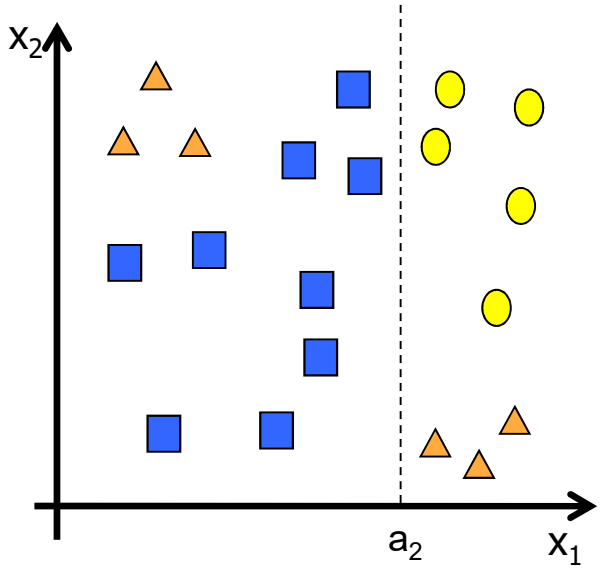
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



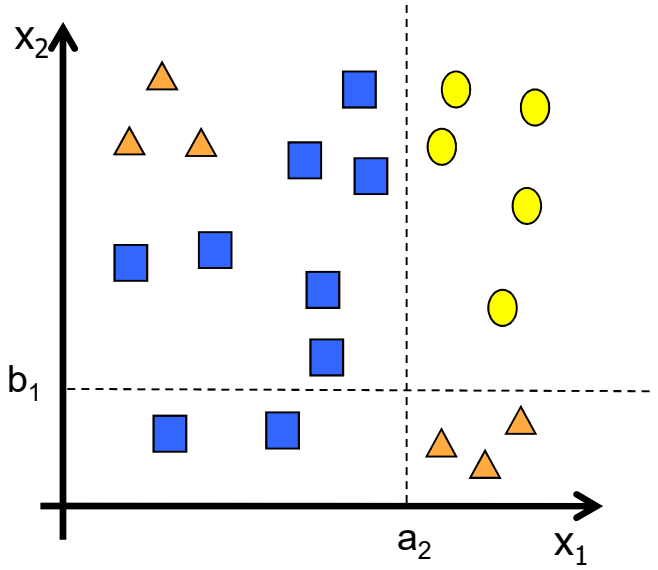
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



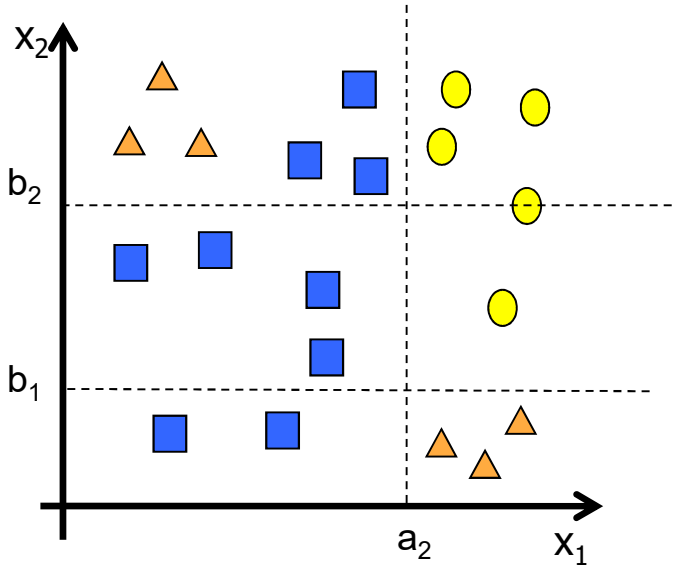
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



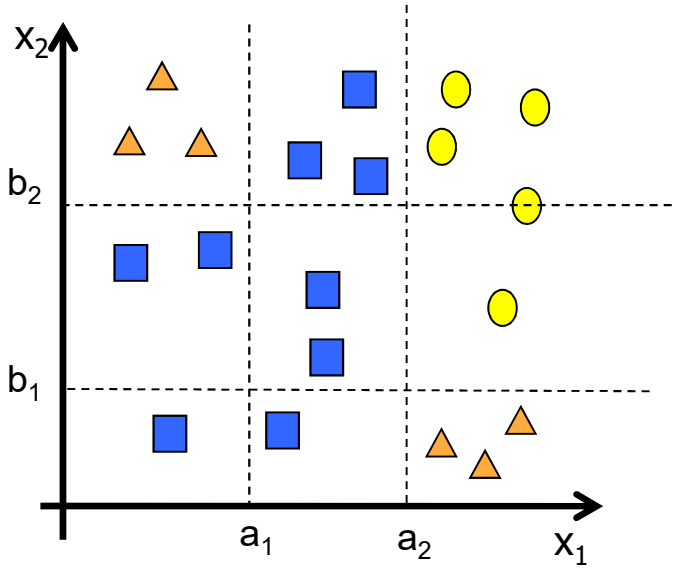
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



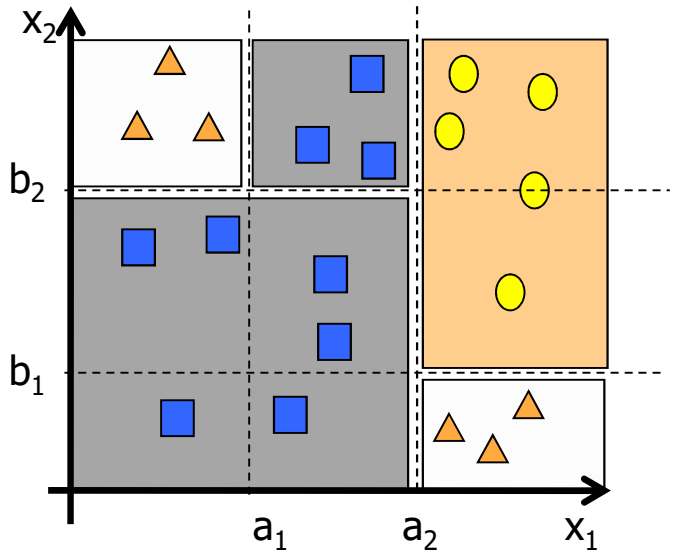
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido

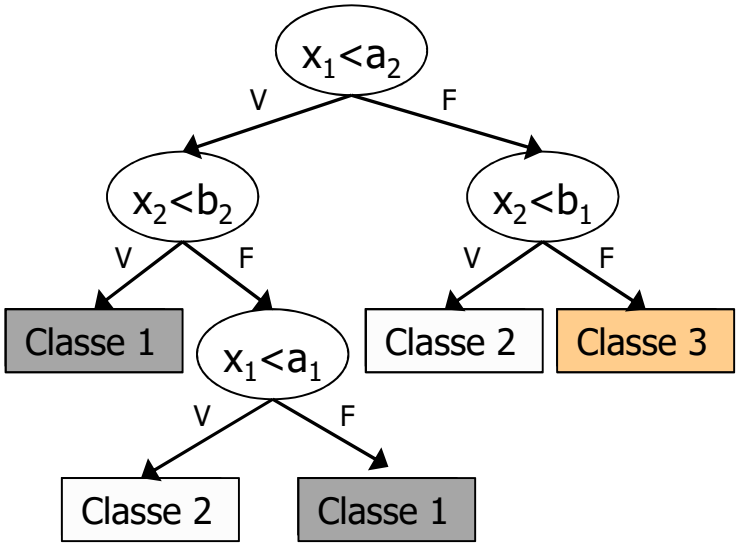


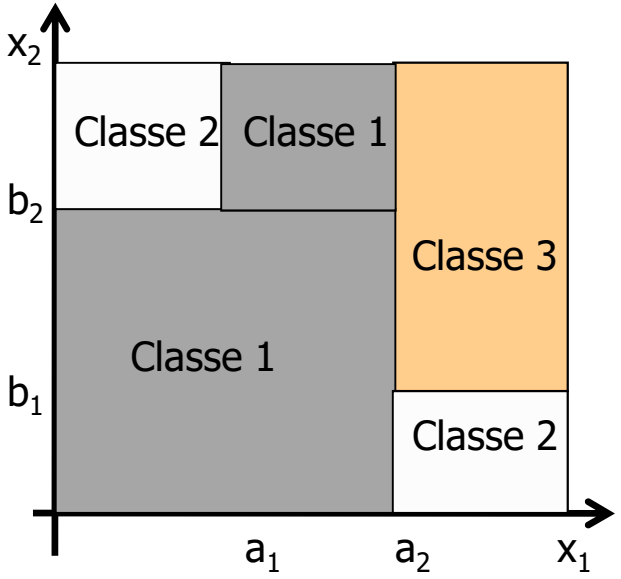
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
    - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



Árvore e partição do espaço de hipóteses





Espaço de hipóteses

- Cada percurso da raiz a um nó folha representa uma regra de classificação
- Cada folha está associada a uma classe
    - Corresponde a um hiper-retângulo no espaço de soluções
        - Cada classe é representada por um conjunto de hiper-retângulos
        - Interseção de hiper-retângulos é um conjunto vazio
        - União de hiper-retângulos cobre todo o espaço

Aspectos positivos das ADs

- Baixo custo de indução e dedução
- Fácil interpretação da hipótese induzida
    - Para árvores pequenas
- Acurácia comparável a de outros classificadores
    - Para conjuntos de dados de baixa complexidade
- Indica atributos preditivos mais relevantes
- Atributos preditivos podem ser numéricos ou simbólicos

Aspectos negativos das ADs

- Dificuldade para predição de valores contínuos
    - Árvores de regressão
- Baixo desempenho em problemas com muitas classes e poucos dados
- Abordagem gulosa
- Limitação de hipóteses a híper-retângulos

Overfitting

- Partição recursiva pode gerar árvores perfeitamente ajustadas aos dados
- Decisões são baseadas em conjuntos cada vez menores de dados
    - Níveis mais profundos podem ter muito poucos dados
        - Presença de ruído nos dados afeta bastante escolha de atributos para esses nós
        - Reduz capacidade de generalização
        - Poda

Poda de árvores

- Elimina parte da árvore
- Pode ser realizada em duas etapas
  - Durante indução (pré-poda)
    - Parar o crescimento da árvore mais cedo
  - Após indução (pós-poda)
    - Crescer a árvore completa e depois podá-la
    - Mais lento, porém mais confiável

Algoritmo C4.5

- Proposto por Quinlan em 1993 como extensão do algoritmo ID3
  - J48
  - C5.0
- Medida de impureza baseada em entropia
- Pós-poda
- Todos os dados precisam caber na memória principal
  - Inadequado para grandes conjuntos de dados

Variações para classificação

- Árvores oblíquas
  - Utiliza uma combinação linear de atributos em cada nó interno
    - Permite fronteiras de decisão oblíquas
- Árvores de opção
  - Cada nó pode ter um conjunto de testes, cada teste para um atributo preditivo
    - Atributos promissores são selecionados

Variações para regressão

- Árvores de regressão
  - Classe de nó folha = média dos valores do atributo alvo dos exemplos que caem nela
    - Utiliza outras medidas para selecionar atributos para nós internos
- Árvores modelo
  - Árvore de regressão combinada com equações de regressão
    - Contêm nas folhas funções de regressão (não) linear

Conclusão

- Árvores de decisão
- Algoritmo de Hunt
- Medidas para escolha de atributos
- Ponto de referência
- Critério de parada
- Espaço de hipóteses

Fim do apresentação