

APRENDIZADO DE ÁRVORES DE DECISÃO

Prof. Igor da Penha Natal

Aprendizagem de Árvore de Decisão

2

- A representação usada é uma árvore de decisão, com um viés para árvores de decisão simples.
- Faz busca no espaço de árvores de decisão, das árvores simples para as mais complexas.
- Toma como entrada um **objeto** ou **situação** descrito por um conjunto de **ATRIBUTOS** (propriedades).
- Retorna uma **decisão** – o valor de saída previsto de acordo com a entrada.
- Os atributos de entrada e o valor de saída podem ser discretos ou contínuos.

Árvore de Decisão

3

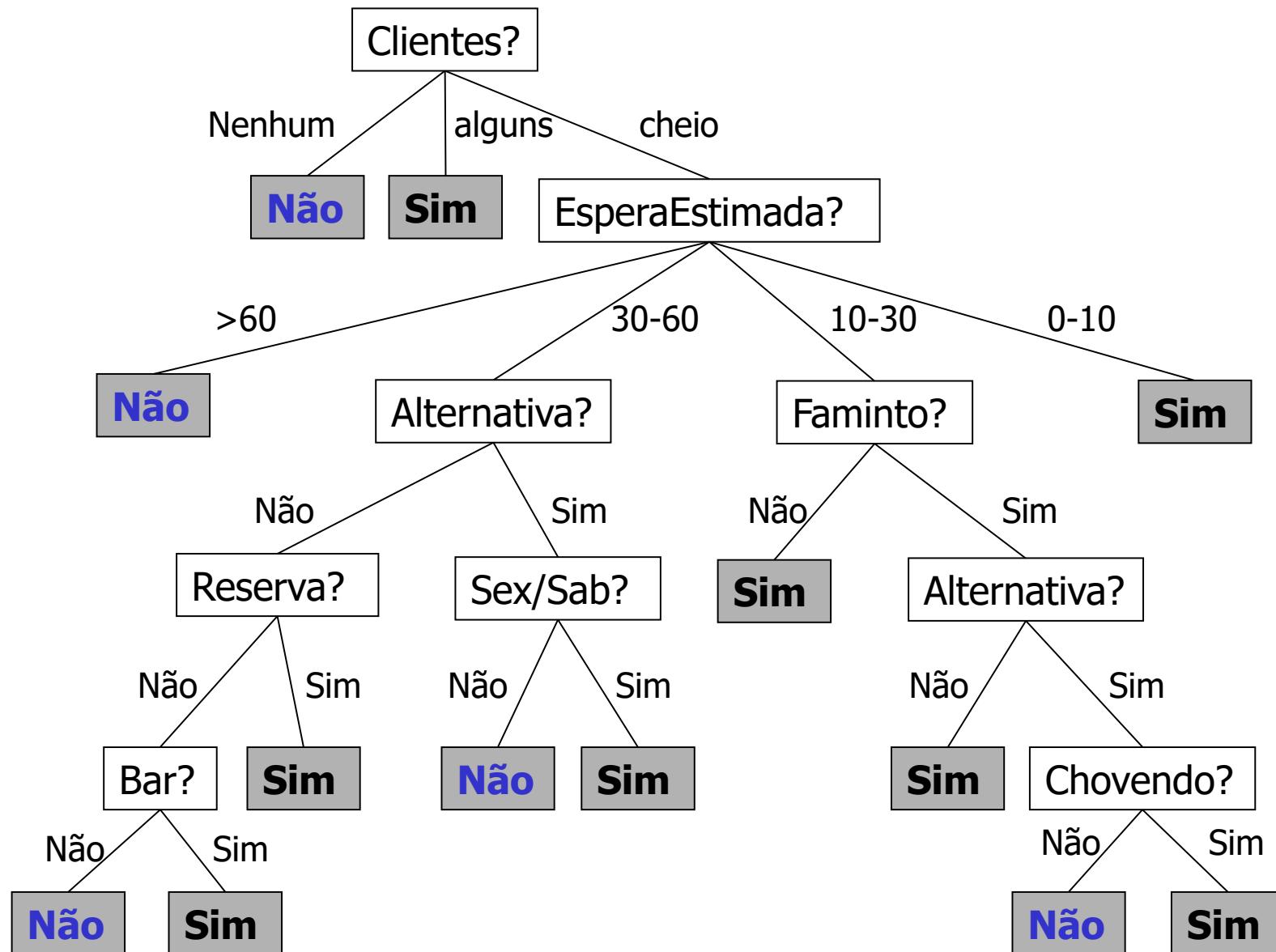
- É uma árvore na qual cada nó **não folha** é rotulado com um **ATRIBUTO DE ENTRADA**.
- Os **arcos** que saem de um nó com o **atributo A** são rotulados com cada possível **VALOR DO ATRIBUTO A**.
- As **folhas** da árvore são rotuladas com a **PREVISÃO PONTUAL DO ATRIBUTO DE SAÍDA**.
- Obtém uma decisão executando uma sequência de testes.
- Cada nó não folha da árvore corresponde a um teste do valor de um atributo.

Exemplo de árvore de decisão

4

- Esperar ou não uma mesa em um restaurante?
- Objetivo: aprender uma definição para o **predicado de objetivo** **vaiEsperar** (booleano).
- Atributos disponíveis:
 - **Alternativa**: há um outro restaurante apropriado por perto?
 - **Bar**: o restaurante tem uma área de bar confortável para esperar?
 - **Sex/Sab**: hoje é sextas ou sábados?
 - **Faminto**: estamos com fome?
 - **Clientes**: quantas pessoas estão no restaurante?
 - **Preço**: a faixa de preços do restaurante.
 - **Chovendo**: está chovendo do lado de fora?
 - **Reserva**: fizemos uma reserva?
 - **Tipo**: qual o tipo do restaurante?
 - **Espera estimada**: o tempo de espera estimado pelo gerente.

Árvore de decisão para o problema do restaurante - Russel



Problemas na aprendizagem de árvore de decisão

- Tendo em conta alguns exemplos de treinamento, que árvore de decisão deve ser gerada?
- Uma árvore de decisão pode representar qualquer função discreta dos atributos de entrada.
- Você precisa de um viés. Por exemplo, preferir a menor árvore. Com menos profundidade? Com menor número de nós? Quais as árvores são as melhores previsoras de dados não vistos ainda?
- Como você deve construir uma árvore de decisão? O espaço de árvores de decisão é demasiado grande para a pesquisa sistemática pela menor árvore de decisão.

Expressividade de árvores de decisão

7

- Qualquer hipótese de árvore de decisão específica para o predicado objetivo **VaiEsperar** pode ser vista como uma asserção da forma:

$$\forall s \text{ } \textit{vaiEsperar}(s) \Leftrightarrow (P_1(s) \vee P_2(s) \vee \dots \vee P_n(s))$$

- Onde cada condição $P_i(s)$ é uma conjunção de testes que correspondem a um caminho da raiz até uma folha da árvore com resultado positivo.

(clientes = alguns) \vee

(clientes = cheio \wedge esperaEstimada = 0 - 10) \vee

(clientes = cheio \wedge esperaEstimada = 10 - 30 \wedge faminto = não) \vee

(clientes = cheio \wedge esperaEstimada = 10 - 30 \wedge faminto = sim \wedge alternativa = não) \vee

(clientes = cheio \wedge esperaEstimada = 10 - 30 \wedge faminto = sim \wedge alternativa = sim \wedge chovendo = sim) \vee

(clientes = cheio \wedge esperaEstimada = 30 - 60 \wedge alternativa = sim \wedge sex/sab = sim) \vee

(clientes = cheio \wedge esperaEstimada = 30 - 60 \wedge alternativa = não \wedge reserva = sim) \vee

(clientes = cheio \wedge esperaEstimada = 30 - 60 \wedge alternativa = não \wedge reserva = não \wedge bar = sim)

Expressividade de árvores de decisão

8

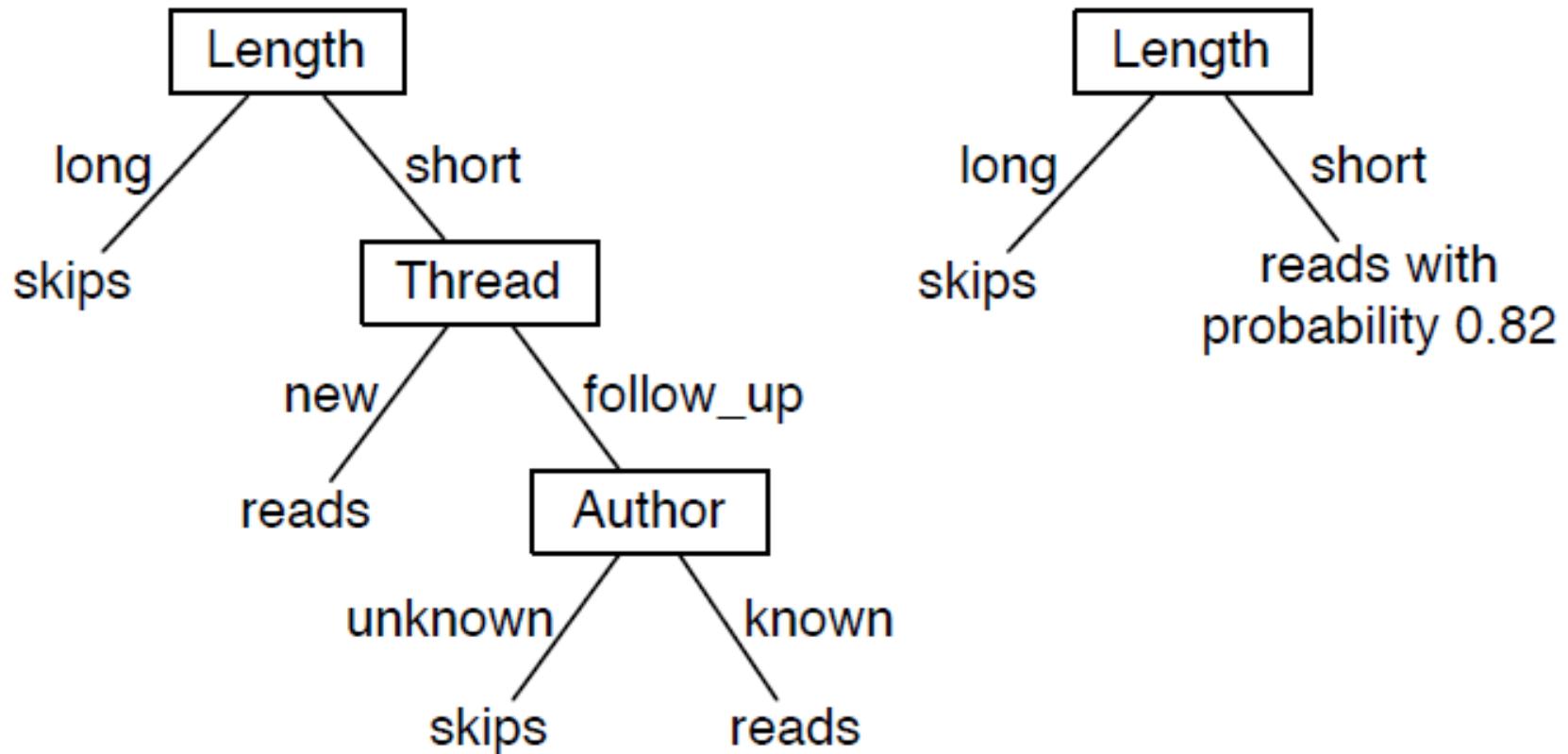
- Uma árvore de decisão:
 - Descreve um relacionamento entre o predicado objetivo e alguma combinação lógica de atributos.
- Árvores de decisão não podem representar testes que se referem a dois ou mais objetos diferentes.

$$\exists r_2 \ Perto(r_2, r) \wedge Preço(r, p) \wedge Preço(r_2, p_2) \wedge MaisBarato(p_2, p)$$

Existe um restaurante mais barato por perto?

- Árvores de decisão são completamente expressivas dentro da classe de linguagens proposicionais.
 - Qualquer função booleana pode ser escrita como uma árvore de decisão.

Exemplo de árvore de decisão



Programa lógico equivalente

skips \leftarrow *long*.

reads \leftarrow *short* \wedge *new*.

reads \leftarrow *short* \wedge *follow-up* \wedge *known*.

skips \leftarrow *short* \wedge *follow-up* \wedge *unknown*.

Expressividade de árvores de decisão

11

- Para qualquer função booleana podemos fazer com que cada linha da Tabela Verdade seja igual a um caminho na árvore.
 - Representação exponencialmente grande.
 - Tabela Verdade tem um número exponencial de linhas.
- Árvores de decisão podem representar muitas funções com árvores muito menores.
- Árvores de decisão servem para alguns tipos de funções e não são boas para outros:
 - **Exemplo:** Função paridade e Função maioria – árvores de decisão exponencialmente grande no tamanho da entrada.
- Infelizmente não existe uma espécie de representação que seja eficiente para todos os tipos de funções.

Problemas apropriados para o uso de árvores de decisão

12

- Instâncias são representadas por meio de pares atributo-valor.
 - Cada atributo assume um número pequeno de possíveis valores disjuntos (por exemplo, Temperatura = Quente, Moderado, Frio).
- A função tem valores discretos (V/F, 1/2/3).
- Os dados de treinamento permitem descrições disjuntas.
- Os dados de treinamento podem conter erros.
- Os dados de treinamento podem conter valores de atributos indefinidos.

Aplicação de árvores de decisão

13

- Muitos problemas práticos possuem as características para aplicação de árvores de decisão.
- Aprendizado utilizando árvore de decisão já foi aplicado a problemas como:
 - Classificar os pacientes médicos pela doença.
 - Causa de maus funcionamentos de equipamentos.
 - A probabilidade de candidatos a empréstimo ficarem inadimplentes.
 - Entre outros...

Induzindo árvores de decisão

14

- Induzindo árvores de decisão por meio de exemplos.
 - Um **EXEMPLO** é descrito pelo **valor de seus atributos** e o **valor do predicado objetivo**.
 - **Valor do predicado objetivo** → classificação do exemplo.
 - Predicado objetivo = verdadeiro → exemplo positivo.
 - Predicado objetivo = falso → exemplo negativo.
 - Um conjunto completo de exemplos é chamado de **conjunto de treinamento**.

Induzindo árvores de decisão

15

- Encontrar uma $AD(h(x))$ que concorde com o conjunto de treinamento parece difícil, mas pode ser trivial.
 - Podemos construir uma **AD** que tem um caminho para uma folha correspondente a cada exemplo.
 - Infelizmente ela não terá muito a informar sobre qualquer outros casos (não fará boas generalizações).
 - Ela memoriza as observações, mas não extrai qualquer padrão.

O problema com a AD trivial

16

- Pela Navalha de Ockham devemos encontrar a menor **AD** consistente com os exemplos (**intratável**).
 - Podemos resolver o problema intratável utilizando algumas heurísticas para encontrarmos árvores pequenas.
- Algoritmo de aprendizagem em árvore de decisão
 - A ideia é testar o **atributo mais importante** – aquele que faz a maior diferença para a classificação dos exemplos.
 - Obter a classificação correta com um pequeno número de testes → caminhos curtos e árvore pequenas.

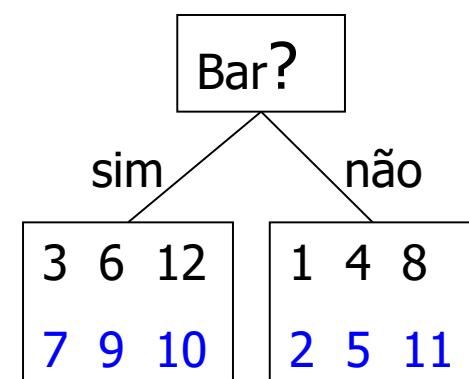
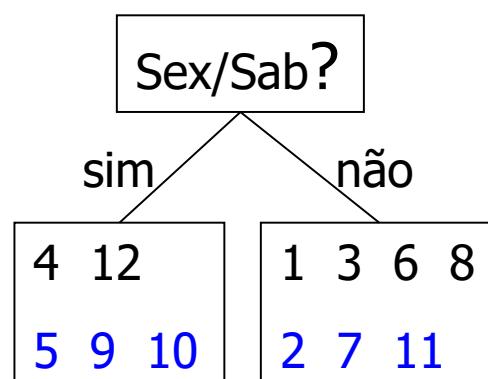
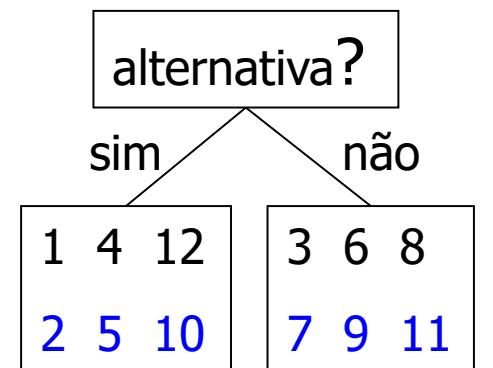
Conjunto de treinamento para o exemplo do restaurante

Ex	Alt	Bar	Sex	Fam	Cli	Pre	Chu	Res	Tipo	Tem	Meta
X_1	Sim	Não	Não	Sim	Alg	\$\$\$	Não	Sim	Fra	0-10	Sim
X_2	Sim	Não	Não	Sim	Che	\$	Não	Não	Tai	30-60	Não
X_3	Não	Sim	Não	Não	Alg	\$	Não	Não	Ham	0-10	Sim
X_4	Sim	Não	Sim	Sim	Che	\$	Sim	Não	Tai	10-30	Sim
X_5	Sim	Não	Sim	Não	Che	\$\$\$	Não	Sim	Fra	>60	Não
X_6	Não	Sim	Não	Sim	Alg	\$\$	Sim	Sim	Ita	0-10	Sim
X_7	Não	Sim	Não	Não	Ne	\$	Sim	Não	Ham	0-10	Não
X_8	Não	Não	Não	Sim	Alg	\$\$	Sim	Sim	Tai	0-10	Sim
X_9	Não	Sim	Sim	Não	Che	\$	Sim	Não	Ham	>60	Não
X_{10}	Sim	Sim	Sim	Sim	Che	\$\$\$	Não	Sim	Ita	10-30	Não
X_{11}	Não	Não	Não	Não	Ne	\$	Não	Não	Tai	0-10	Não
X_{12}	Sim	Sim	Sim	Sim	Che	\$	Não	Não	Ham	30-60	Sim

Aplicação do algoritmo de aprendizagem

18

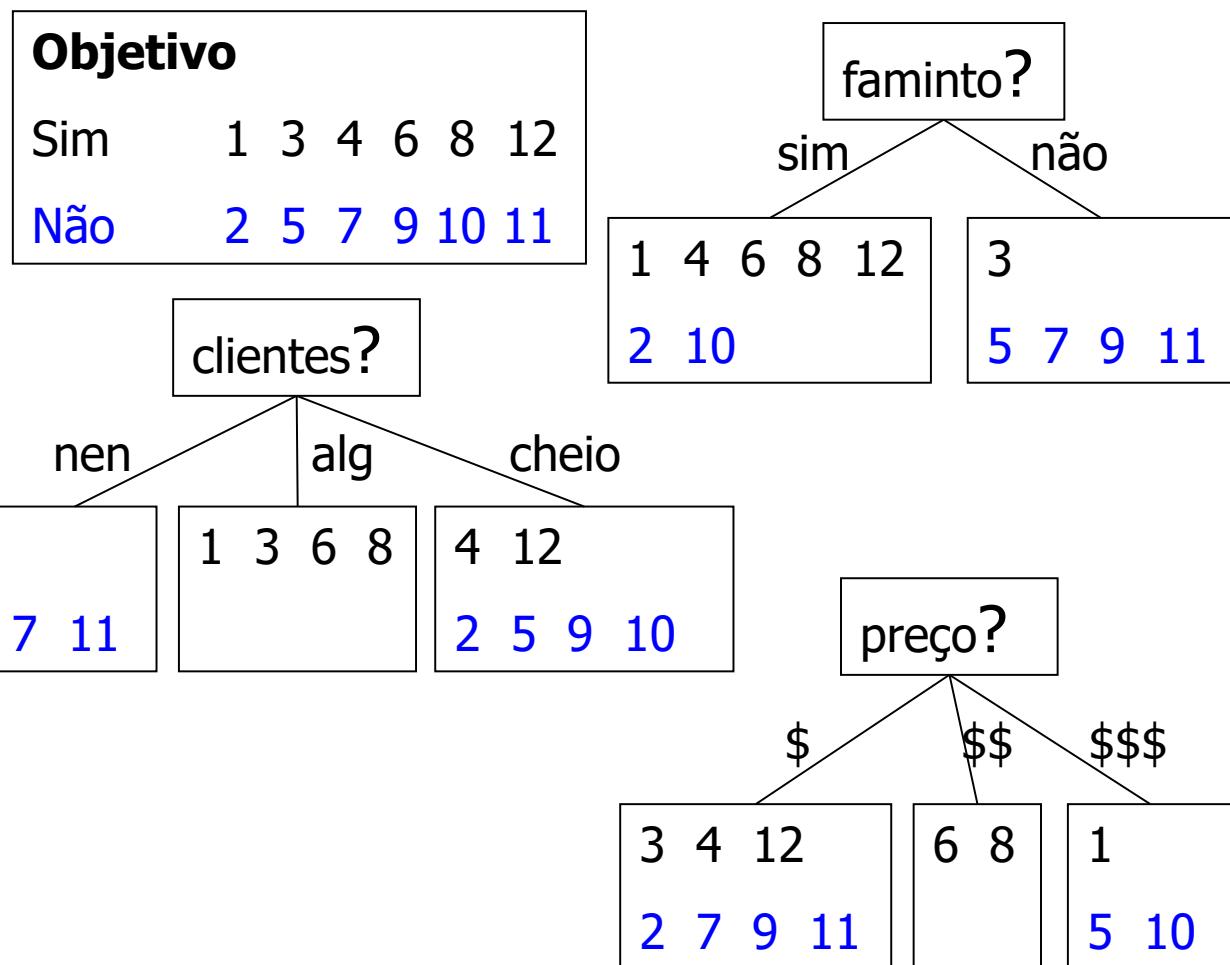
Ex	Alt	Bar	Sex
X_1	Sim	Não	Não
X_2	Sim	Não	Não
X_3	Não	Sim	Não
X_4	Sim	Não	Sim
X_5	Sim	Não	Sim
X_6	Não	Sim	Não
X_7	Não	Sim	Não
X_8	Não	Não	Não
X_9	Não	Sim	Sim
X_{10}	Sim	Sim	Sim
X_{11}	Não	Não	Não
X_{12}	Sim	Sim	Sim



Aplicação do algoritmo de aprendizagem

19

Ex	Fam	Cli	Pre
X_1	Sim	Alg	\$\$\$
X_2	Sim	Che	\$
X_3	Não	Alg	\$
X_4	Sim	Che	\$
X_5	Não	Che	\$\$\$
X_6	Sim	Alg	\$\$
X_7	Não	Ne	\$
X_8	Sim	Alg	\$\$
X_9	Não	Che	\$
X_{10}	Sim	Che	\$\$\$
X_{11}	Não	Ne	\$
X_{12}	Sim	Che	\$



Exercício

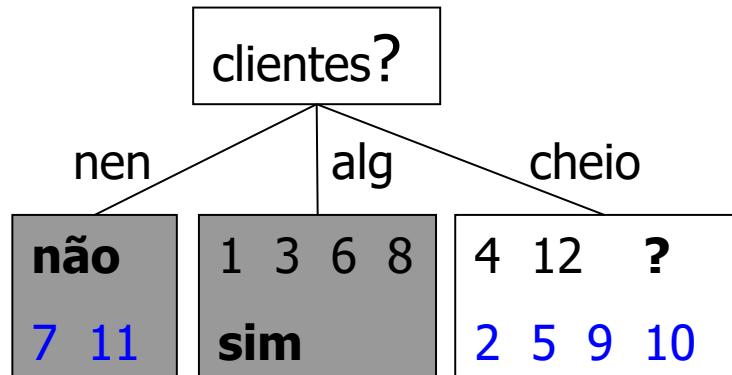
20

- Fazer a representação dos atributos abaixo da forma especificada anteriormente.
 - Chovendo?
 - Reserva?
 - Tipo?
 - Tempo de espera?
- Qual é o atributo que separa melhor os exemplos?

Qual é o atributo que melhor separa os exemplos?

21

- O quanto cada atributo classifica:
 - **Alternativa, Bar, Sex/Sab, Faminto, Chovendo, Reserva, Tipo:** zero exemplos.
 - **Clientes:** 6 exemplos.
 - **Preço e Espera estimada:** 2 exemplos.
- O atributo ***cliente*** classifica 6 exemplos.
- Então ele será a raiz da árvore de decisão:



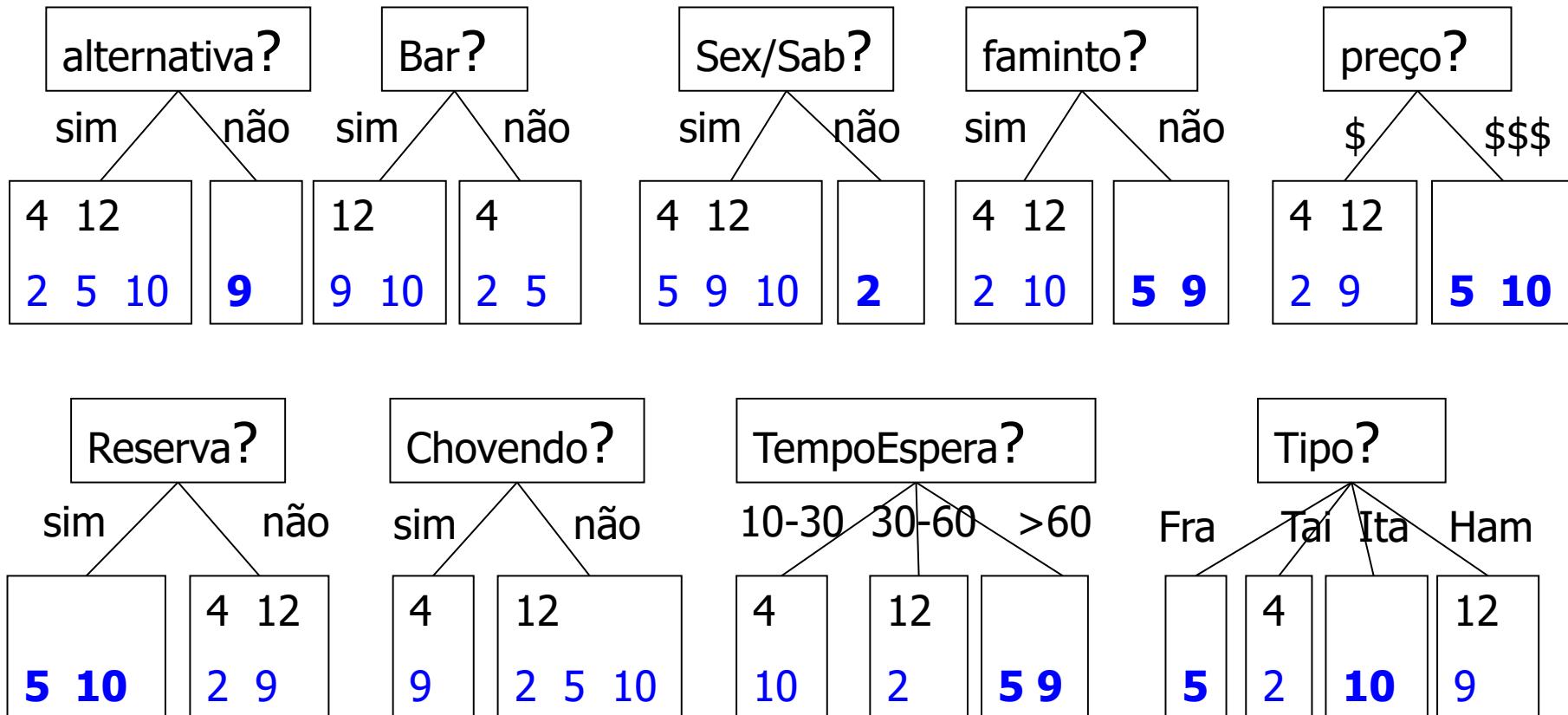
Escolha dos atributos

22

- Depois da escolha do atributo ***clientes*** ficamos com um conjunto misto de exemplos se o valor for cheio.
- Depois que o primeiro teste de atributo separa os exemplos, cada resultado é um novo problema de aprendizagem em **AD**.
 - Com menos exemplos (os que ainda não foram classificados - 2 5 9 10 4 e 12) e um atributo a menos (tira-se *clientes*).
- **Exercício**: Aplique o algoritmo de aprendizagem de **AD** para os exemplos anteriores.

Nova instância do problema com menos exemplos e menos atributos

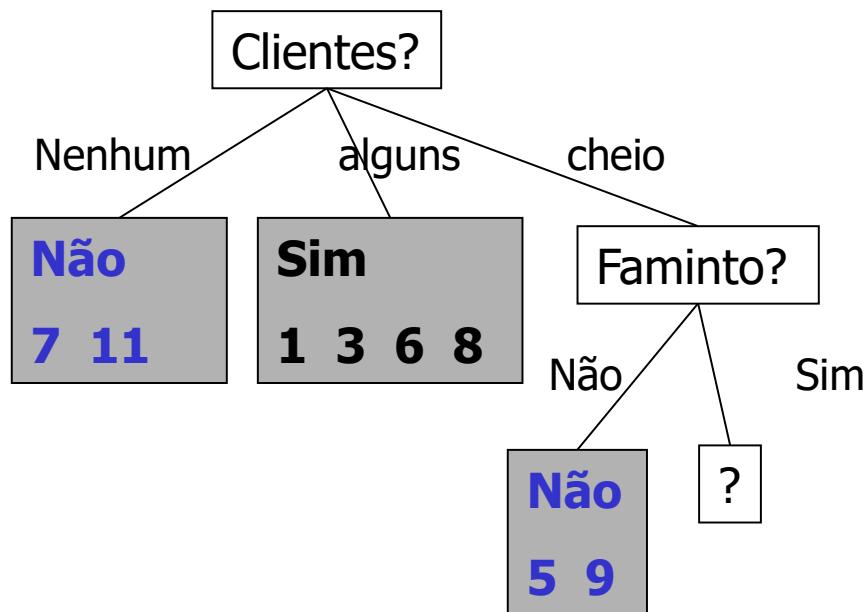
23



Qual é o atributo que melhor separa os exemplos restantes?

24

- Faminto, preço, reserva e tipo classificam 2 exemplos.
 - Podemos escolher qualquer um deles (heurística gulosa).



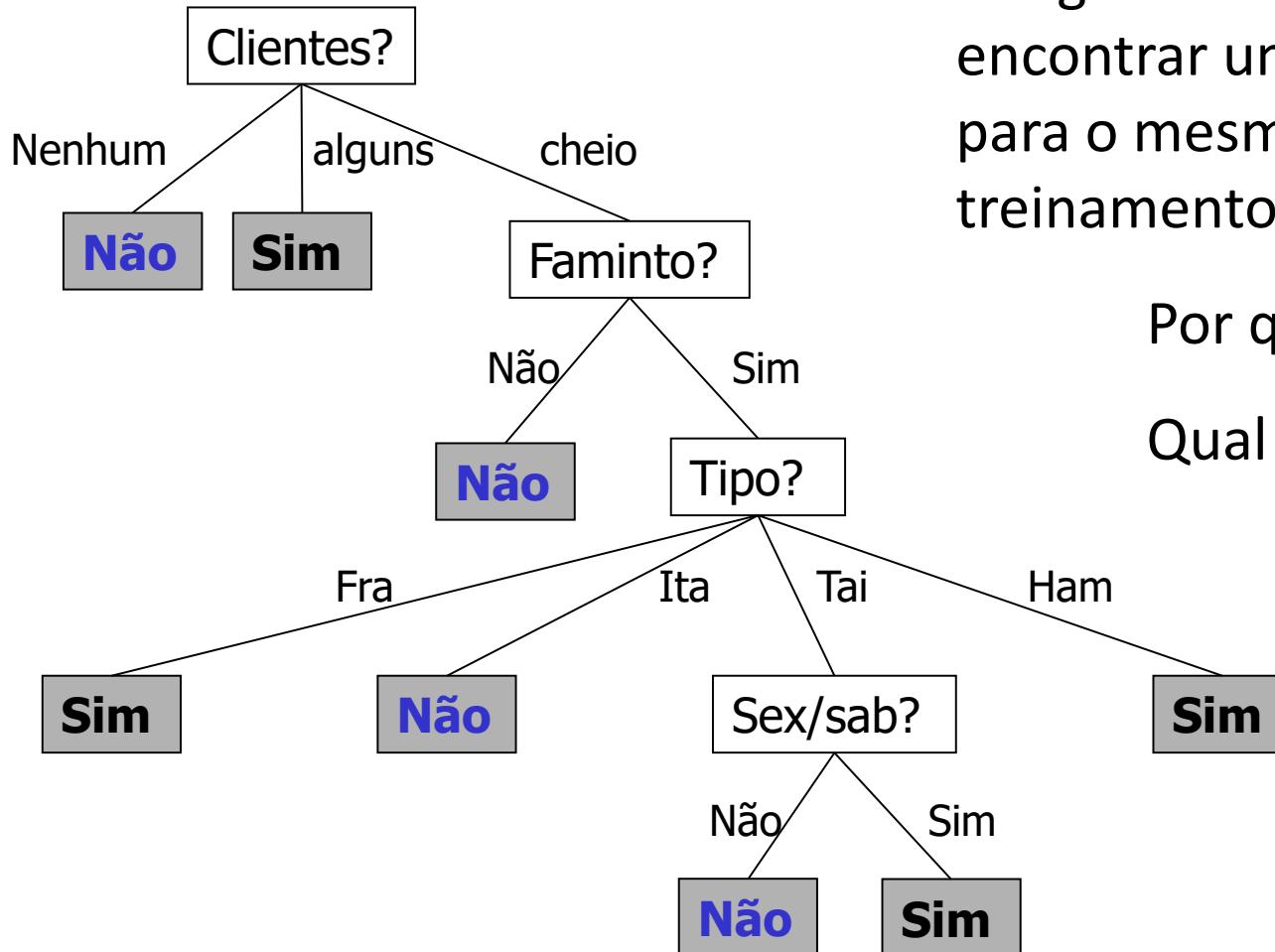
Casos a considerar na escolha de um novo atributo

25

- Se existem alguns exemplos positivos e alguns negativos, escolha o melhor atributo para dividi-los.
- Se todos os exemplos restantes forem positivos (ou negativos), terminamos.
- Se não resta nenhum atributo, mas há exemplos positivos e negativos – descrições iguais com classificações diferentes = **RUÍDO** nos dados.

Árvore resultante da aplicação do algoritmo ID3

27



O algoritmo poderia encontrar uma AD diferente para o mesmo conjunto de treinamento?

Por que?

Qual seria?

A árvore resultante

28

- É diferente da árvore original.
- Mas a hipótese concorda com todos os exemplos e é consideravelmente mais simples do que a árvore original.
- *Chovendo* e *Reserva* ficaram de fora porque a árvore não necessita deles para classificar os exemplos.
- Se tivéssemos um conjunto de treinamento maior, com certeza a árvore ficaria mais parecida com a original.

Escolha de testes de atributos

29

- Esquema de aprendizagem da ***AD***:
 - Projetado para minimizar a profundidade da árvore final.
 - **IDEIA**: escolher o atributo que melhor fornece uma classificação exata dos exemplos.
- **Atributo perfeito**: divide os exemplos em conjuntos que são todos positivos ou todos negativos.
 - *Clientes* não é perfeito, mas é “bastante bom”.
- **Atributo inútil**: deixa os conjuntos de exemplos com a mesma proporção do conjunto original.
 - *Tipo* é um atributo “realmente inútil”.

Quantidade de informação fornecida pelo atributo

30

- Como medimos um “atributo muito bom” ou um “atributo realmente inútil”?
- A medida:
 - Deve ter o valor máximo quando o atributo é perfeito.
 - E o valor mínimo quando o atributo for completamente inútil.
- Uma medida apropriada seria a **QUANTIDADE ESPERADA DE INFORMAÇÃO** fornecida pelo atributo.
 - Isto é feito por meio da teoria da informação.

Entropia

31

- Medida comumente usada na teoria da informação.
- Caracteriza a **impureza** de uma coleção de exemplos.
- Seja S uma coleção de exemplos positivos e negativos de algum conceito objetivo lógico, a entropia de S :

$$\text{Entropia} \equiv - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

onde p_{\oplus} é a proporção de exemplos positivos em S

e p_{\ominus} é a proporção de exemplos negativos em S

Cálculo de Entropia – Exemplo

32

- Suponha que S seja uma coleção de 14 exemplos de algum conceito lógico:
 - 9 exemplos são positivos e 5 são negativos [9+, 5-]

$$\text{Entropia}([9+, 5-]) \equiv -(9/14) \log_2(9/14) - (5/14) \log_2(5/14)$$

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

- Entropia=0, se todos os exemplos de S pertencem à mesma classe (todos positivos ou todos negativos).
- Entropia=1, quando S contém um número igual de exemplos positivos e negativos (exemplo do restaurante).
- Se S contém números desiguais de exemplos positivos e negativos a entropia está entre 0 e 1.

Medida de Ganho de informação

33

- A medida da efetividade de um atributo para classificar os dados de treinamento
- **GANHO DE INFORMAÇÃO** = redução esperada da entropia causada pelo particionamento dos exemplos de S por um atributo A

$$\text{Ganho}(S, A) = \text{Entropia}(S) - \sum_{v \in \text{valores}(A)} \left(\frac{|S_v|}{|S|} \right) \text{Entropia}(S_v)$$

onde :

- $\text{valores}(A)$ é o conjunto de todos os possíveis valores para o atributo A .
- S_v é o subconjunto de S para o qual o atributo A tem valor v .

Medida de Ganho de informação – Exemplo para o atributo *Cientes*

34

$$S = [6+, 6-] \quad S_{nenhum} = [0+, 2-] \quad S_{alguns} = [4+, 0-] \quad S_{cheio} = [2+, 4-]$$

$$\text{Ganho}(S, \text{Cientes}) = \text{Entropia}(S) - \sum_{v \in (\text{nenhum, alguns, cheio})} (S_v / S) \text{Entropia}(S_v)$$

$$\begin{aligned} \text{Ganho}(S, \text{Cientes}) &= \text{Entropia}(S) - (\frac{2}{12}) \text{Entropia}(S_{nenhum}) - (\frac{4}{12}) \text{Entropia}(S_{alguns}) \\ &\quad - (\frac{6}{12}) \text{Entropia}(S_{cheio}) \end{aligned}$$

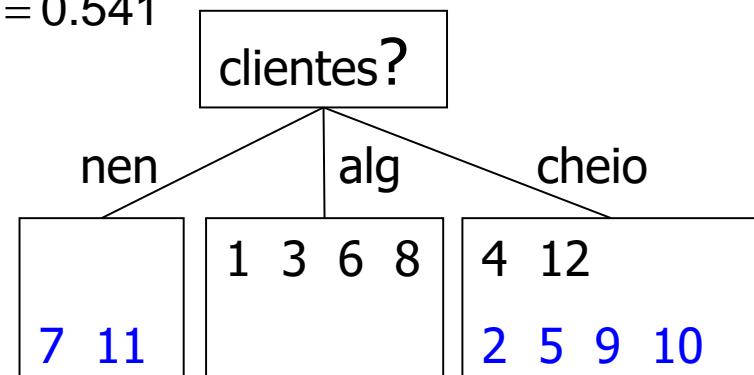
$$\text{Entropia}(S) = -(\frac{2}{12})\log_2(\frac{2}{12}) - (\frac{2}{12})\log_2(\frac{2}{12}) = 1 \text{ (# de positivos e negativos é igual)}$$

$$\text{Entropia}(S_{nenhum}) = -(\frac{2}{2})\log_2(\frac{2}{2}) - (\frac{2}{2})\log_2(\frac{2}{2}) = 0 \text{ (todos negativos)}$$

$$\text{Entropia}(S_{alguns}) = -(\frac{4}{4})\log_2(\frac{4}{4}) - (\frac{4}{4})\log_2(\frac{4}{4}) = 0 \text{ (todos positivos)}$$

$$\text{Entropia}(S_{cheio}) = -(\frac{6}{6})\log_2(\frac{6}{6}) - (\frac{6}{6})\log_2(\frac{6}{6}) = 0.918$$

$$\text{Ganho}(S, \text{Cientes}) = 1 - (\frac{2}{12}) 0 - (\frac{4}{12}) 0 - (\frac{6}{12}) 0.918 = 0.541$$



Medida de Ganho de informação – Exemplo para o atributo *Tipo*

35

$$S = [6+, 6-]$$

$$S_{\text{Francês}} = [1+, 1-] \quad S_{\text{Italiano}} = [1+, 1-]$$

$$S_{\text{Tailandês}} = [2+, 2-] \quad S_{\text{Hamburger}} = [2+, 2-]$$

$$\text{Ganho}(S, \text{Tipo}) = \text{Entropia}(S) - \sum_{v \in (\text{Francês, Italiano, Tailandês, Hamburger})} \left(\frac{S_v}{S} \right) \text{Entropia}(S_v)$$

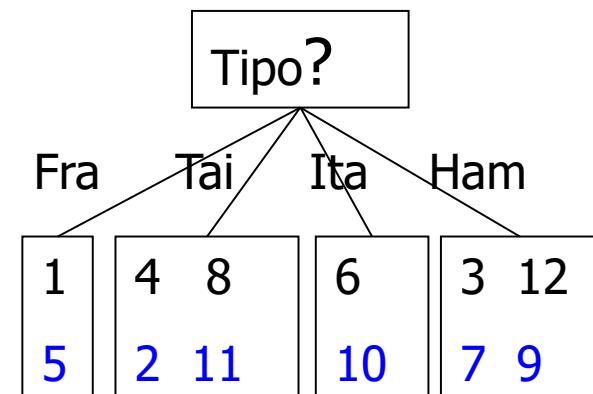
$$\begin{aligned} \text{Ganho}(S, \text{Tipo}) &= \text{Entropia}(S) - \left(\frac{2}{12} \right) \text{Entropia}(S_{\text{Fra}}) - \left(\frac{2}{12} \right) \text{Entropia}(S_{\text{Ita}}) \\ &\quad - \left(\frac{4}{12} \right) \text{Entropia}(S_{\text{Tai}}) - \left(\frac{4}{12} \right) \text{Entropia}(S_{\text{Ham}}) \end{aligned}$$

$\text{Entropia}(S) = 1$ (# de exemplos positivos e negativos é igual)

$$\text{Entropia}(S_{\text{Fra}}) = \text{Entropia}(S_{\text{Ita}}) = \text{Entropia}(S_{\text{Tai}}) = \text{Entropia}(S_{\text{Ham}}) = 1$$

(em todos os subconjuntos o número de exemplos positivos e negativos é igual)

$$\text{Ganho}(S, \text{Tipo}) = 1 - \left(\frac{2}{12} \right) 1 - \left(\frac{2}{12} \right) 1 - \left(\frac{4}{12} \right) 1 - \left(\frac{4}{12} \right) 1 = 0$$



Conclusões sobre a Medida de Ganho de Informação

36

- O atributo *Cientes* é melhor do que o atributo *Tipo*.
 - *Cientes* tem maior ganho de informação para separar os exemplos.
 - *Tipo* é um atributo inútil para classificar os exemplos
 - $\text{Ganho}(S, \text{Tipo}) = 0$
- O ganho de informação deve ser calculado para todos os atributos e o conjunto S de exemplos.
 - *Cientes* será o atributo com maior ganho.
- Após a escolha de *Cientes* como a raiz da árvore, o novo conjunto S se torna $S_{cheio}[2+, 4-]$ e $\text{Entropia}(S_{cheio})=0.918$
 - Novamente o ganho de informação deve ser calculado para todos os atributos e o novo conjunto S .

Exercício

37

- Calcule o ganho de informação para os outros atributos do restaurante, considerando que o atributo *Clientes* já foi escolhido como raiz da árvore.
- Continue o algoritmo de aprendizagem calculando o ganho de informação para os atributos, para provar a árvore de decisão que foi induzida no slide 24.

Como avaliar um algoritmo de aprendizagem?

38

- É um bom algoritmo se ele produz hipóteses que classificam (predizem) bem exemplos ainda não vistos.
- A **PREDIÇÃO** é boa se ela se torna verdadeira.
- Como avaliar a qualidade de uma hipótese?
 - Podemos checar sua previsão com uma classificação correta que já conhecemos.
 - Conjunto de exemplos conhecidos = **conjunto de testes**.

Metodologia para avaliação de desempenho de um algoritmo

39

- Coletar um grande conjunto de exemplos.
- Dividi-lo em dois conjuntos disjuntos:
 - Conjunto de treinamento e conjunto de teste
- Aplicar o algoritmo ao conjunto de treinamento, gerando uma hipótese h .
- Medir a quantidade de exemplos do conjunto de teste classificados corretamente por h .
- Repetir as etapas 1 a 4 para:
 - Diferentes tamanhos de conjuntos de treinamento.
 - Diferentes conjuntos de treinamento de cada tamanho.

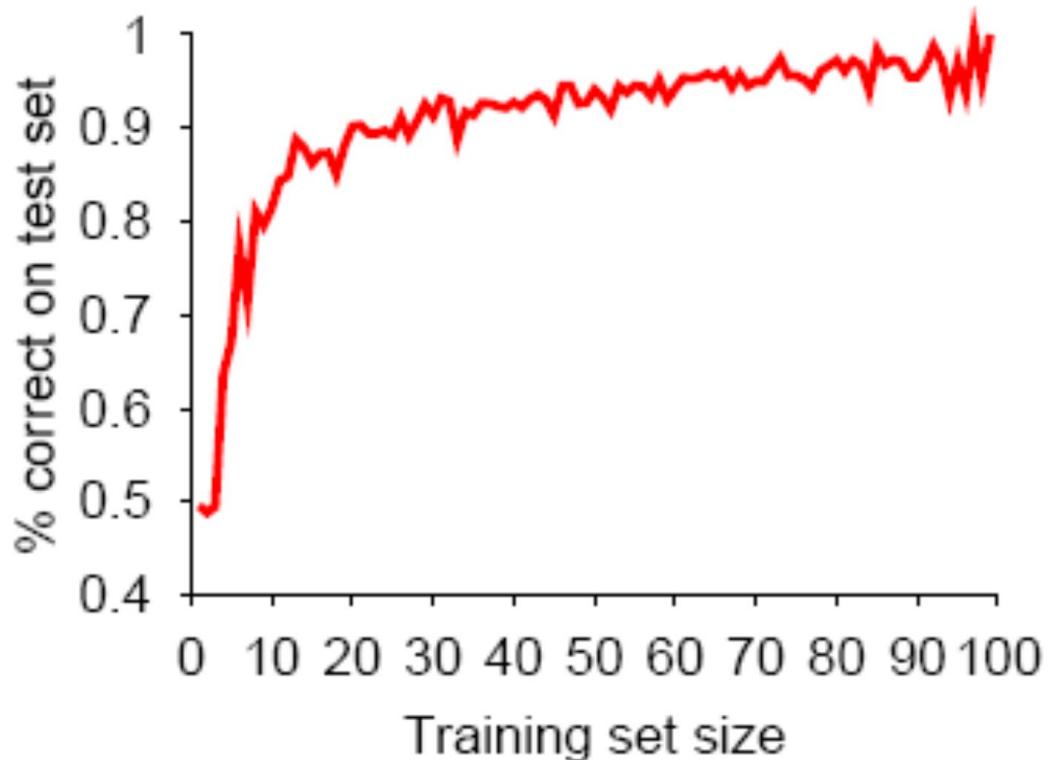
Curva de aprendizagem

40

- É traçada com o conjunto de dados obtidos da metodologia anterior.
- Conjunto de treinamento aumenta → qualidade da previsão aumenta.
- Bom sinal de que existe um padrão nos dados e o algoritmo está capturando este padrão.

Curva de aprendizagem

41



Ruído e superadaptação (*overfitting*)

42

- O algoritmo ID3 faz crescer cada ramo da árvore o suficiente para classificar perfeitamente os exemplos de treino.
- **Problemas:**
 - Quando existe **ruídos** ou **erros aleatórios** nos dados, ou
 - Quando o **número de exemplos de treino é muito pequeno** não constituindo uma amostra representativa da verdadeira função objetivo.
 - Nestes casos ID3 pode produzir árvores que se **superadaptam** aos exemplos de treino - isto é, aprendem inclusive os ruídos e os erros.

Definição de superadaptação

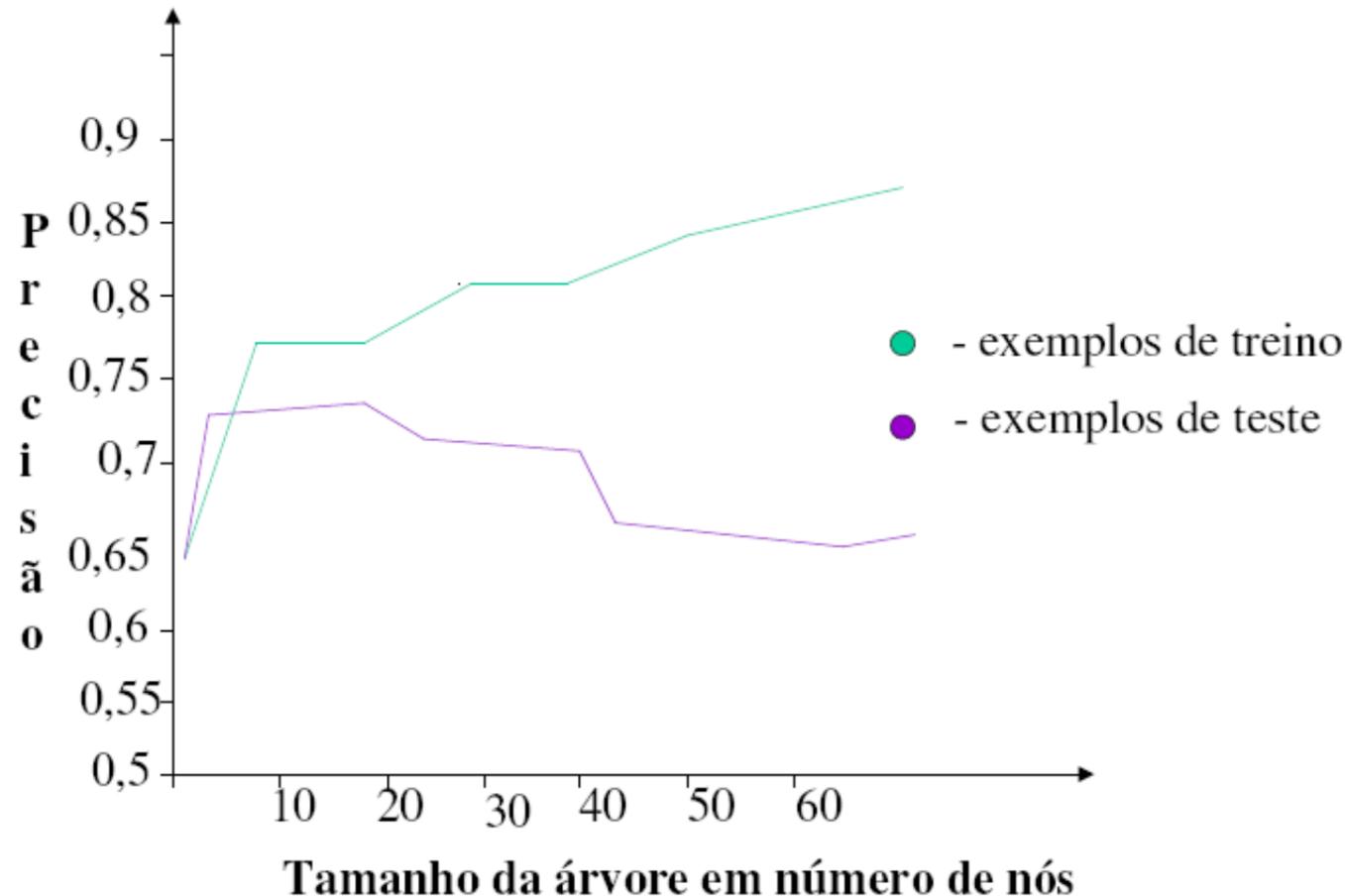
43

Dado um espaço de hipóteses \mathbf{H} . Uma hipótese $\mathbf{h} \in \mathbf{H}$ overfits os dados de treino se existe alguma hipótese alternativa $\mathbf{h}' \in \mathbf{H}$, tal que \mathbf{h} tenha menor erro do que \mathbf{h}' sobre os exemplos de treino, mas \mathbf{h}' tem menor erro do que \mathbf{h} sobre toda a distribuição dos exemplos (i.e. incluindo exemplos fora do conjunto de treinamento)

- A superadaptação aflige todo tipo de algoritmo de aprendizagem, não apenas algoritmos para árvores de decisão.

Exemplo de superadaptação

44



Como evitar a superadaptação

45

- Fazer a árvore parar de crescer antes que ela alcance o ponto onde ela classifique perfeitamente os exemplos de treino.
 - Para isso temos que acompanhar as porcentagens de previsões corretas tanto no conjunto de teste quanto no conjunto de treinamento.
 - Um atributo com ganho de informação perto de zero tem grandes indícios de ser um atributo irrelevante.
 - Quando a partição dos dados não for estatisticamente significante, podemos parar o crescimento da árvore.

Como evitar a superadaptação

46

- Permitir que a árvore “sobreajuste” os dados, e depois podar os atributos irrelevantes.
 - Podar um nó de decisão: remover a subárvore 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 podada resultante não apresenta um comportamento pior do que a original sobre o conjunto de validação (erro de poda reduzido).

Como evitar a superadaptação

47

- Particionar os dados em conjuntos de validação e treinamento:
 - Faça até que uma redução (poda) adicional seja prejudicial:
 - 1. Avaliar o impacto sobre o conjunto de validação da poda de cada nó possível, mais aqueles abaixo dele.
 - 2. Remover “gulosamente” aquele que melhora mais a precisão sobre o conjunto de validação.

Como evitar a superadaptação

48

- A **VALIDAÇÃO CRUZADA** é outra técnica que reduz a superadaptação.
- Ela pode ser aplicada a qualquer algoritmo de aprendizagem.
 - A ideia é estimar o quanto cada hipótese irá estimar dados não vistos.
 - Separa-se uma fração dos dados conhecidos como dados de teste e induz-se a hipótese com o restante dos dados.
 - A validação cruzada de k vias significa que deve-se realizar k experimentos reservando cada vez um fração $1/k$ diferente dos dados para testes e calcular a média dos resultados.
 - Valores populares de k são 5 e 10. O extremo $k = n$ também é conhecido como validação cruzada com omissão de um.

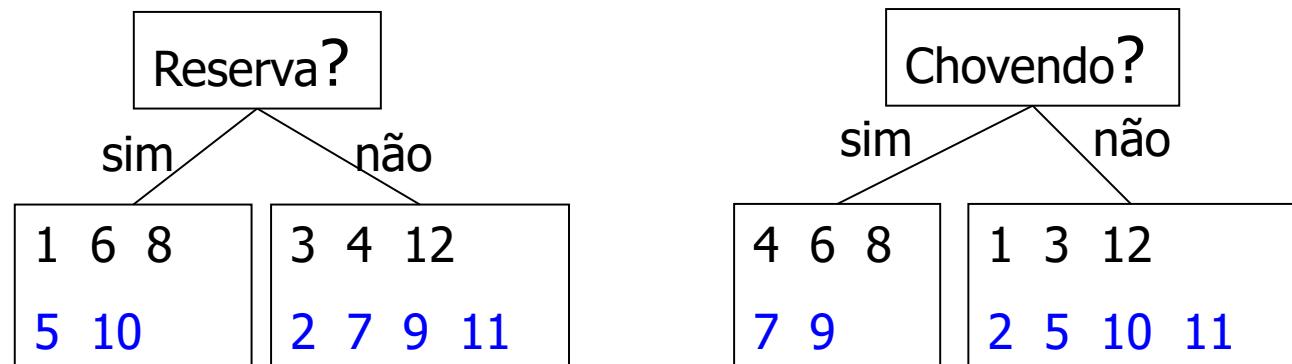
Aplicação do algoritmo de aprendizagem

49

Ex	Cho	Res
X_1	Não	Sim
X_2	Não	Não
X_3	Não	Não
X_4	Sim	Não
X_5	Não	Sim
X_6	Sim	Sim
X_7	Sim	Não
X_8	Sim	Sim
X_9	Sim	Não
X_{10}	Não	Sim
X_{11}	Não	Não
X_{12}	Não	Não

Meta

Sim	1	3	4	6	8	12
Não	2	5	7	9	10	11



Aplicação do algoritmo de aprendizagem

50

