Pontifícia Universidade Católica
do Rio de Janeiro

INF 1771 – Inteligência Artificial

Aula 12 – Árvores de Decisão

2016.1

Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>

Árvores de Decisão

- Uma das formas de algoritmo de aprendizado mais **simples** e de **maior sucesso**.

- Uma árvore de decisão tem como entrada um objeto ou situação descritos por um **conjunto de atributos** e como saída uma "**decisão**" (previsão do valor de saída dada a entrada).

- Uma árvore de decisão toma as suas decisões através de uma sequência de testes.

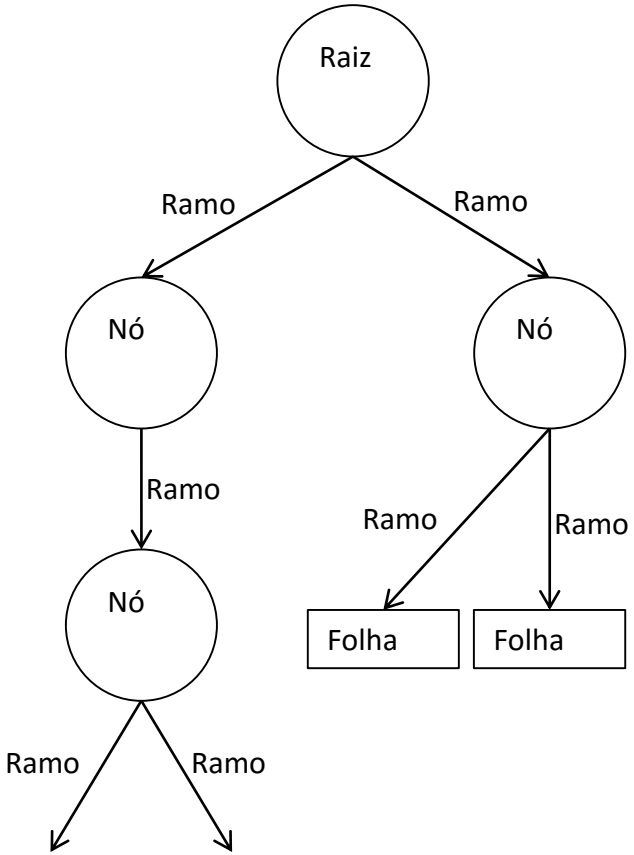
Árvores de Decisão

- Uma árvore de decisão utiliza uma estratégia de dividir-para-conquistar:
  - Um problema complexo é decomposto em subproblemas 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 subespaços.
  - A cada sub-espaço é associada uma classe.

Árvores de Decisão

- Cada **nó** interno da árvore corresponde a um teste do valor de uma propriedade.
- Os **ramos** dos nós são rotulados com os resultados possíveis do teste.
- Cada **nó folha** da árvore específica o valor a ser retornado se aquela folha for alcançada.
- A representação de uma árvore de decisão é bem natural para os seres humanos.



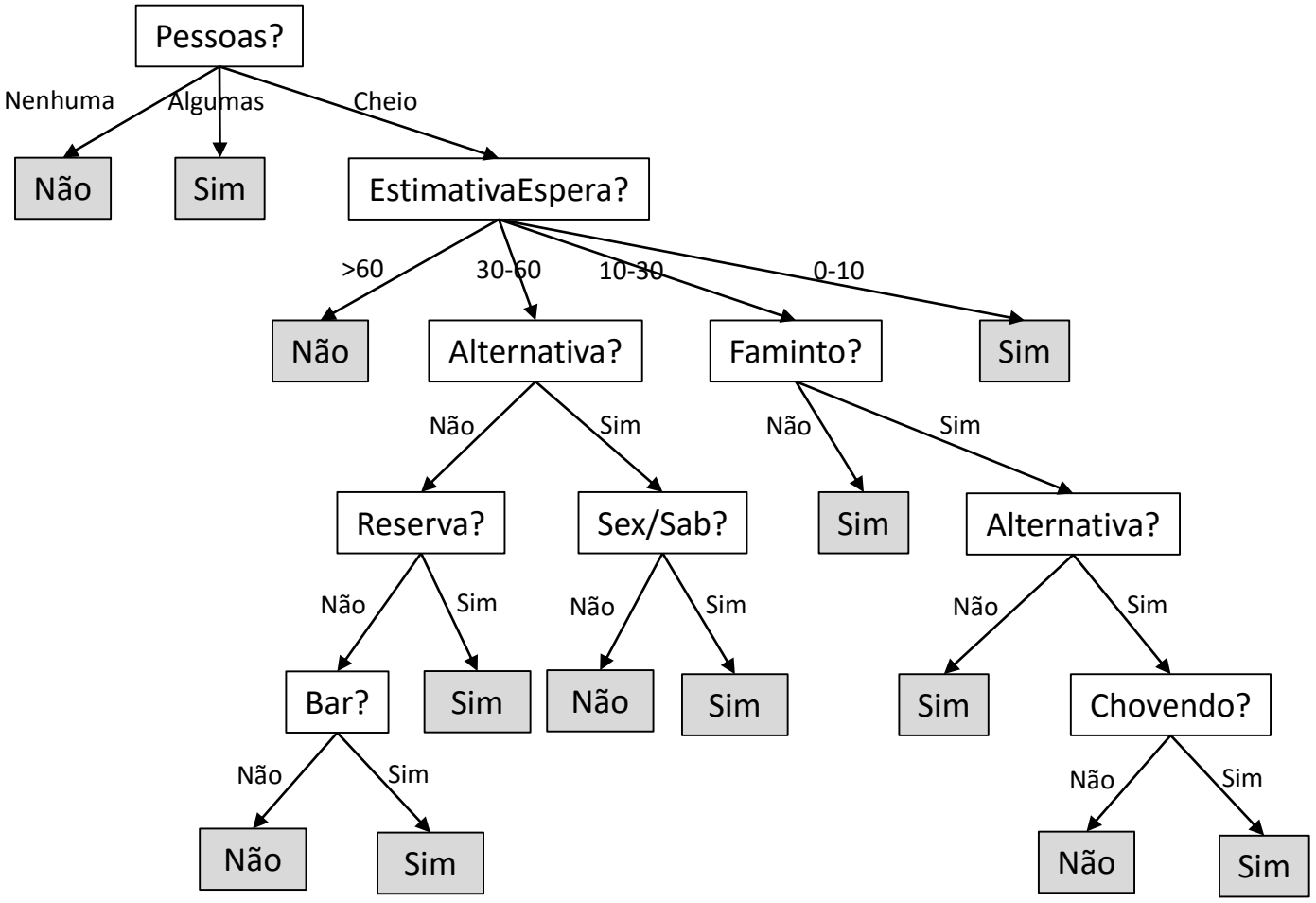
Exemplo – Restaurante

- **Problema:** Esperar por uma mesa em um restaurante.
- O **objetivo** é aprender uma definição para o predicado “vai esperar”.
- Primeiramente é necessário definir quais **atributos** estão disponíveis para descrever alguns exemplos nesse domínio.

Exemplo – Restaurante

- **Atributos:**
  - **Alternativa:** Verdadeiro se existe um restaurante alternativo adequado nas proximidades.
  - **Bar:** Verdadeiro se o restaurante tem uma área de bar confortável para ficar esperando.
  - **Sex/Sab:** Verdadeiro se o dia da semana for sexta ou sábado.
  - **Faminto:** Verdadeiro se estamos com fome.
  - **Pessoas:** Quantas pessoas estão no restaurante (os valores são Nenhuma, Algumas e Cheio).
  - **Preço:** Preço do restaurante de ($, $ $, $$$).
  - **Chuva:** Verdadeiro se está chovendo lá fora.
  - **Reserva:** Verdadeiro se nós fizemos uma reserva.
  - **Tipo:** Tipo de restaurante (Francês, Italiano, Tailandês, Hambúrguer).
  - **EstimativaEspera:** Tempo de espera estimado (00-10, 10-30, 30-60, > 60 minutos).

Exemplo – Restaurante



Gerando Árvores de Decisão a partir de Exemplos

- É possível gerar uma árvore de decisão a partir de um **conjunto de exemplos**.

- **Exemplos positivos** são aqueles que levam a uma resposta positiva.

  Exemplo: “vai esperar” = Sim.

- **Exemplos negativos** são aqueles que levam a uma resposta negativa.

  Exemplo: “vai esperar” = Não.

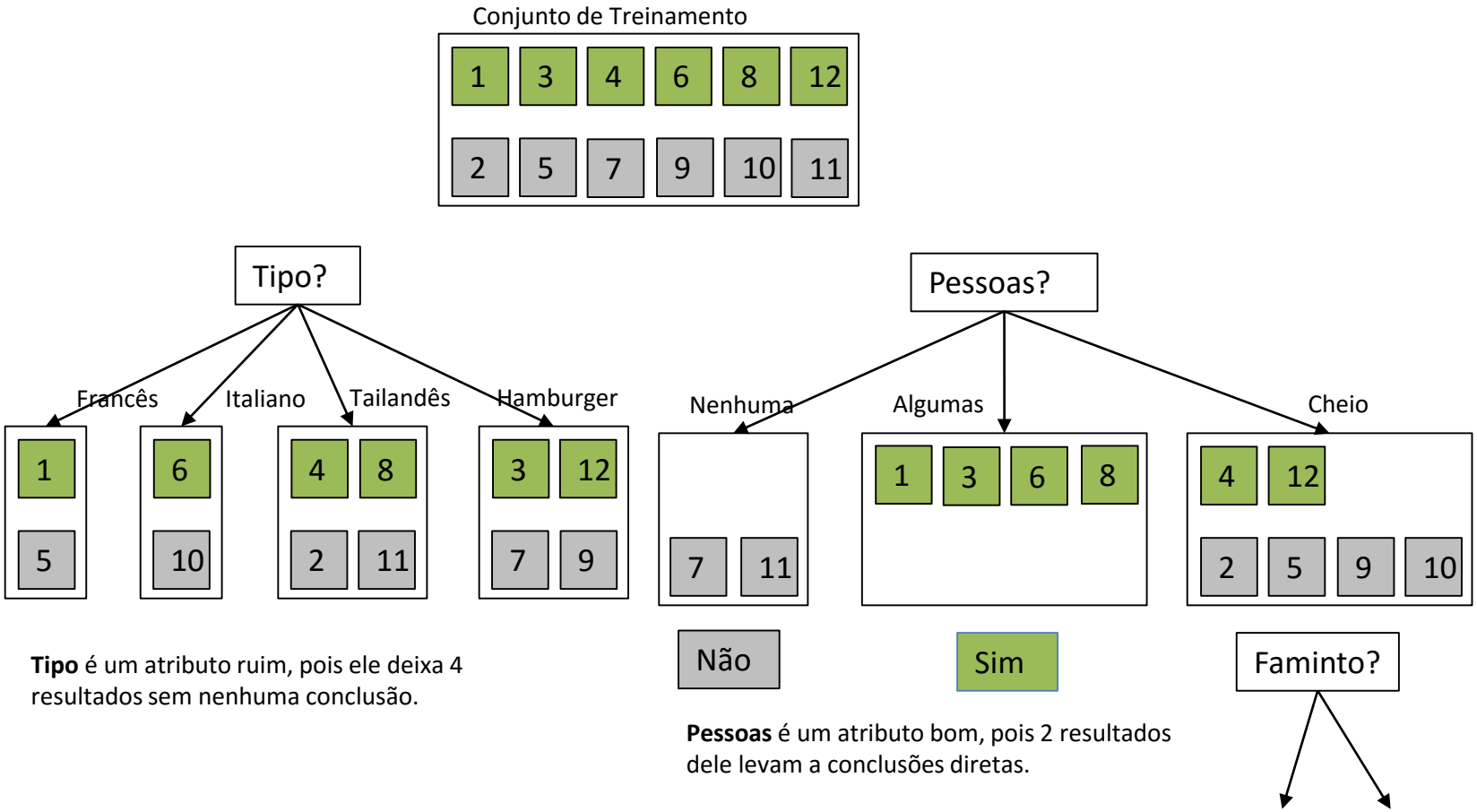
Conjunto de Treinamento

	Atributos										Obj.
Exemplo	**Alt.**	**Bar**	**S/S**	**Fam.**	**Pes.**	**Pre.**	**Chov.**	**Res.**	**Tipo**	**Est.**	**Esp.**
X_1	Sim	Não	Não	Sim	Algumas	$$$	Não	Sim	Fran.	0-10	Sim
X_2	Sim	Não	Não	Sim	Cheio	$	Não	Não	Tai.	30-60	Não
X_3	Não	Sim	Não	Não	Algumas	$	Não	Não	Ham.	0-10	Sim
X_4	Sim	Não	Sim	Sim	Cheio	$	Sim	Não	Tai.	10-30	Sim
X_5	Sim	Não	Sim	Não	Cheio	$$$	Não	Sim	Fran.	>60	Não
X_6	Não	Sim	Não	Sim	Algumas	$$	Sim	Sim	Ital.	0-10	Sim
X_7	Não	Sim	Não	Não	Nenhuma	$	Sim	Não	Ham.	0-10	Não
X_8	Não	Não	Não	Sim	Algumas	$$	Sim	Sim	Tai.	0-10	Sim
X_9	Não	Sim	Sim	Não	Cheio	$	Sim	Não	Ham.	>60	Não
X_{10}	Sim	Sim	Sim	Sim	Cheio	$$$	Não	Sim	Ital.	10-30	Não
X_{11}	Não	Não	Não	Não	Nenhuma	$	Não	Não	Tai.	0-10	Não
X_{12}	Sim	Sim	Sim	Sim	Cheio	$	Não	Não	Ham.	30-60	Sim

Gerando Árvores de Decisão a partir de Exemplos

- Seguindo o **principio de Ockham**, devemos encontrar a menor árvore de decisão que seja consistente com os exemplos de treinamento.
  - "Qualquer fenómeno deve assumir apenas as premissas estritamente necessárias à explicação do fenómeno e eliminar todas as que não causariam qualquer diferença aparente nas predições da hipótese ou teoria."
- A ideia básica do algoritmo é testar os **atributos mais importantes** primeiro.
  - O atributo mais importante é aquele que faz mais diferença para a classificação de um exemplo.
- Dessa forma, esperamos conseguir a classificação correta com um pequeno número de testes.

Gerando Árvores de Decisão a partir de Exemplos



Tipo é um atributo ruim, pois ele deixa 4 resultados sem nenhuma conclusão.

Pessoas é um atributo bom, pois 2 resultados dele levam a conclusões diretas.

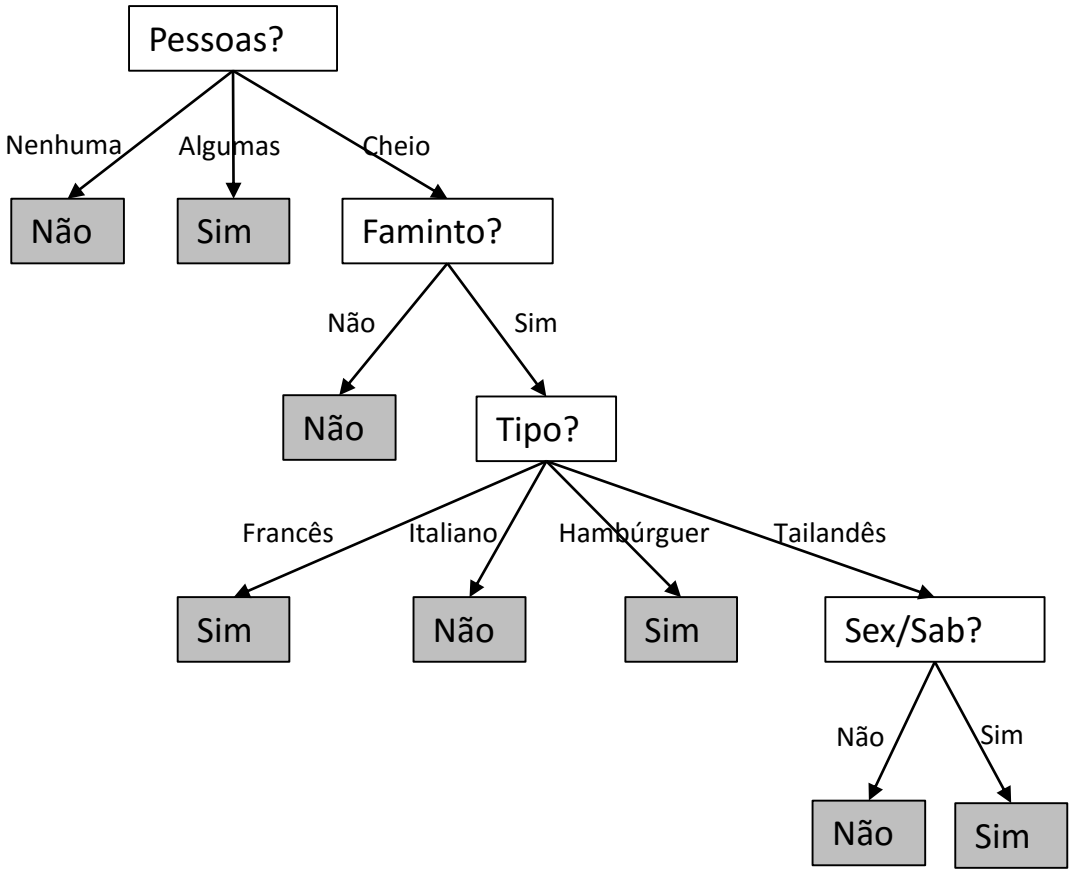
Gerando Árvores de Decisão a partir de Exemplos

- Algoritmo:
  - **(1)** Enquanto existirem exemplos positivos e negativos, deve-se escolher o melhor atributo para dividi-los.
  - **(2)** Se todos os exemplos restantes forem positivos (ou todos negativos), então podemos responder Sim ou Não.
  - **(3)** Se não existirem exemplos restantes, retorna um valor padrão calculado a partir da classificação da maioria dos atributos do nó pai.
  - **(4)** Se não existirem atributo restantes, mas ainda existirem exemplos positivos e negativos temos um problema.

Gerando Árvores de Decisão a partir de Exemplos

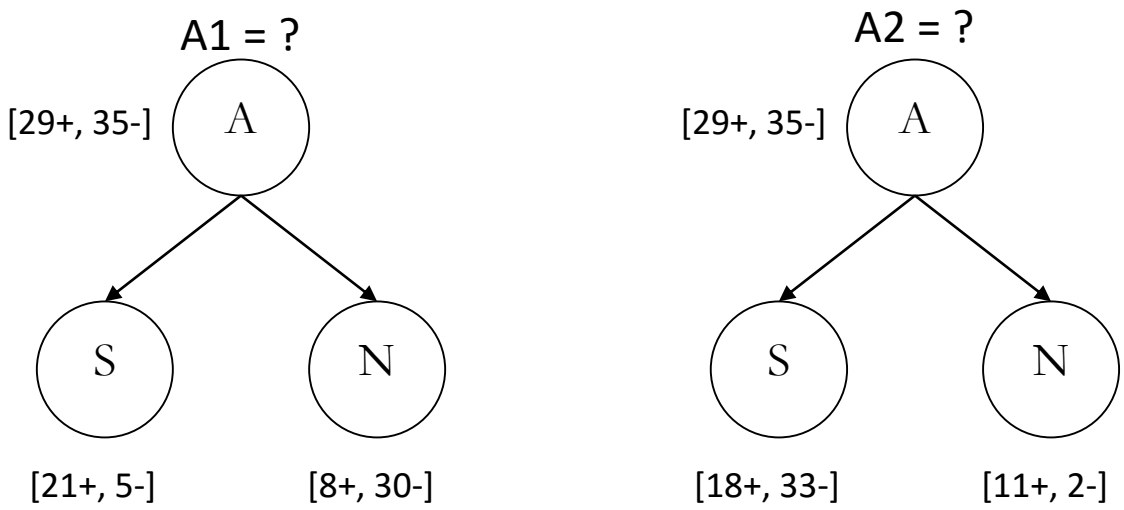
- Quando não existem atributos restantes, mas ainda existem exemplos positivos e negativos significa que:
  - Esses exemplos têm exatamente a **mesma descrição**, mas **classificações diferentes**. Isso acontece quando alguns dos dados estão incorretos, ou seja há **ruído nos dados**.
  - Também acontece quando os atributos **não dão informação suficiente** para descrever a situação completamente, ou quando o domínio é realmente **não-determinístico**.
  - Uma saída simples do problema é a utilização de uma **votação majoritária**.

Gerando Árvores de Decisão a partir de Exemplos



Escolhendo os Melhores Atributos

- Qual é o melhor atributo?



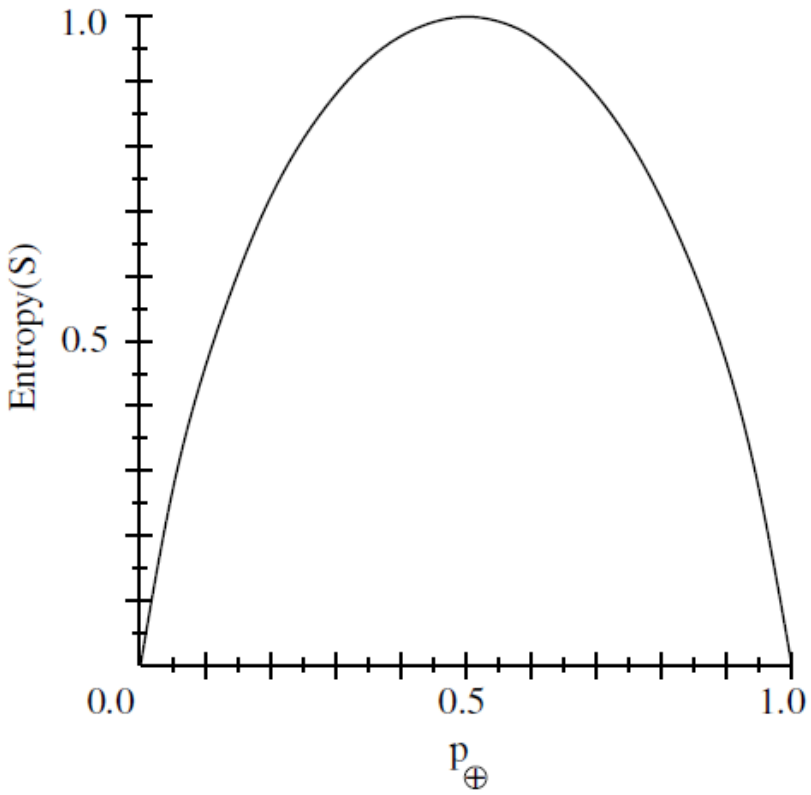
Escolhendo os Melhores Atributos

- **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:

$$\text{Entropia}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

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

Escolhendo os Melhores Atributos



Escolhendo os Melhores Atributos

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

Escolhendo os Melhores Atributos

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

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

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

- A função entropia relativa a uma classificação varia entre 0 e 1.

Escolhendo os Melhores Atributos

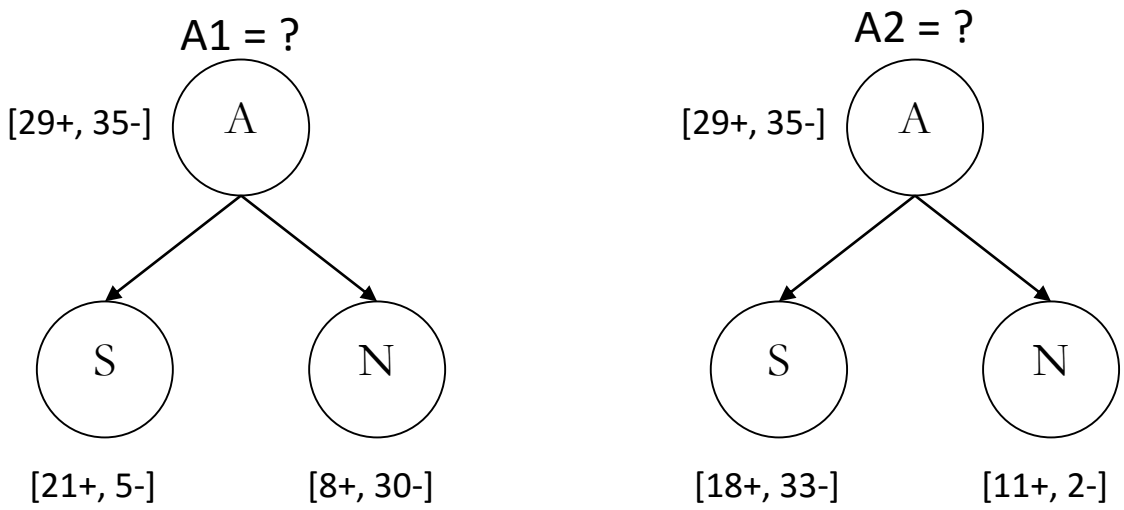
- Generalizando para o caso de um atributo alvo aceitar n diferentes valores, a entropia de S relativa a esta classificação de n–classes é definida como:

$$\text{Entropia(S)} = \sum_{i=1}^{n} - p_i \log_b p_i$$

- Onde b = número de classes (normalmente b = 2)
- Entropia é uma medida da aleatoriedade (impureza) de uma variável.
- A entropia tem máximo ($log_b\ i$) se $p_i = p_j$ para qualquer i ≠ j
- A entropia(x) = 0 se existe um i tal que $p_i = 1$
- É assumido que $0 * log_b\ 0 = 0$

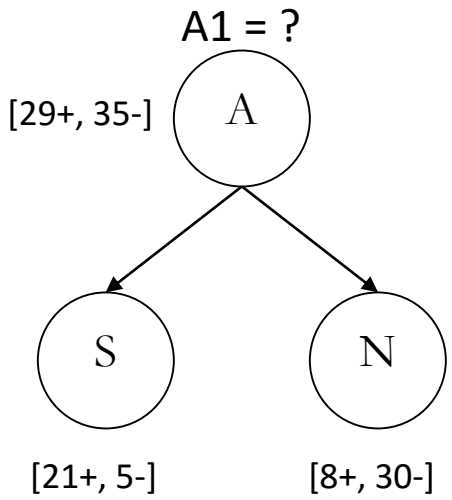
Escolhendo os Melhores Atributos

- Qual é o melhor atributo?



Escolhendo os Melhores Atributos

- Qual é o melhor atributo?



Ent(A) = $-(29/64 \log_2 29/64) - (35/64 \log_2 35/64) = 0.994$

Ent(S) = $-(21/26 \log_2 21/26) - (5/26 \log_2 5/26) = 0.706$

Ent(N) = $-(8/38 \log_2 8/38) - (30/38 \log_2 30/38) = 0.742$

Info(A1) = (Ent(S) * 29/64) + (Ent(N) * 35/64) = 0.726

Ganho(A1) = 0.994 – 0.726 = 0.268

Escolhendo os Melhores Atributos

- Qual é o melhor atributo?

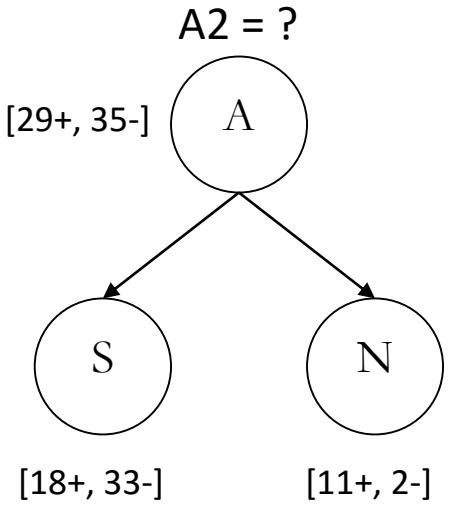
Ent(A) = -(29/64 $\log_2$ 29/64) - (35/64 $\log_2$ 35/64) = 0.994

Ent(S) = -(18/51 $\log_2$ 18/51) - (33/51 $\log_2$ 33/51) = 0.937

Ent(N) = -(11/13 $\log_2$ 11/13) - (2/13 $\log_2$ 2/13) = 0.619

Info(A2) = (Ent(S) * 29/64) + (Ent(N) * 35/64) = 0.763

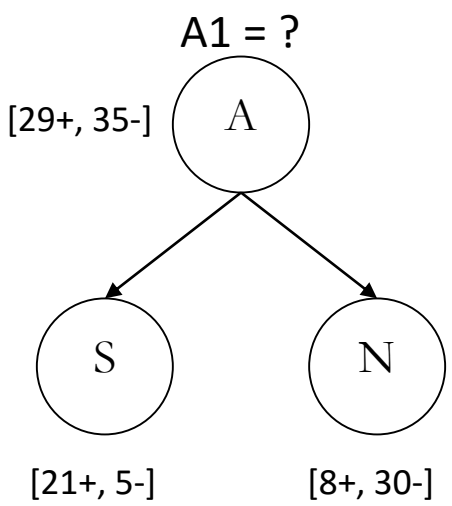
Ganho(A2) = 0.994 – 0.726 = 0.230

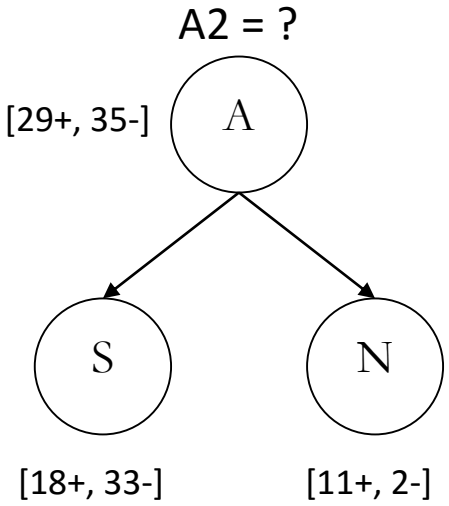


Escolhendo os Melhores Atributos

- Qual é o melhor atributo?

Ent(A) = 0.994





Info(A1) = 0.726

Ganho(A1) = 0.268

Info(A2) = 0.763

Ganho(A2) = 0.230

Exemplo 2 – Partida de Tênis

Tempo	Temperatura	Umidade	Vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	69	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	96	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

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

Tempo	Joga
Sol	Não
Sol	Não
Sol	Não
Sol	Sim
Sol	Sim

Cálculo do Ganho de Informação de um Atributo Nominal

- 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

Joga	Sol	Nublado	Chuva
Sim	2	4	3
Não	3	0	2

- 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

Temperatura	Joga
85	Não
80	Não
83	Sim
70	Sim
69	Sim
65	Não
64	Sim
72	Não
69	Sim
75	Sim
75	Sim
72	Sim
81	Sim
71	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

- Como medir o ganho de informação desta partição?

Joga	T > =70.5	T < 70.5
Sim	5	4
Não	4	1

- 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 | temperatura < 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

Medindo Desempenho

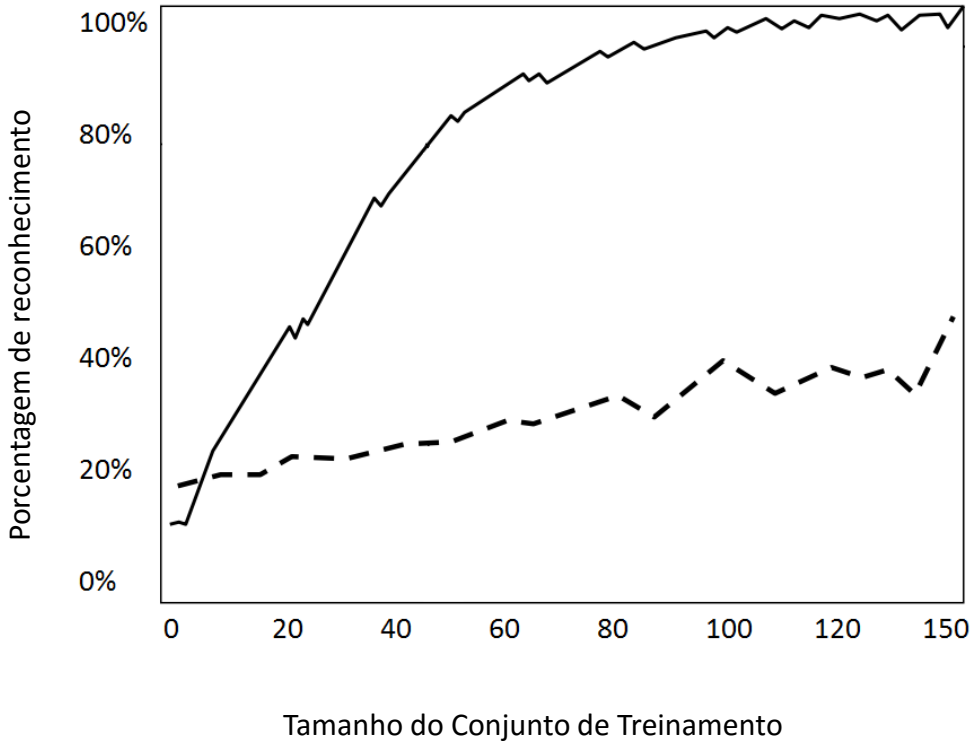
- Um algoritmo de aprendizado **é bom** se ele produz hipóteses que conseguem **prever a classificação** de exemplos **não vistos**.

- A maneira mais simples de se medir o desempenho de um método de aprendizado é realizando a classificação de um conjunto de **exemplos de teste**.

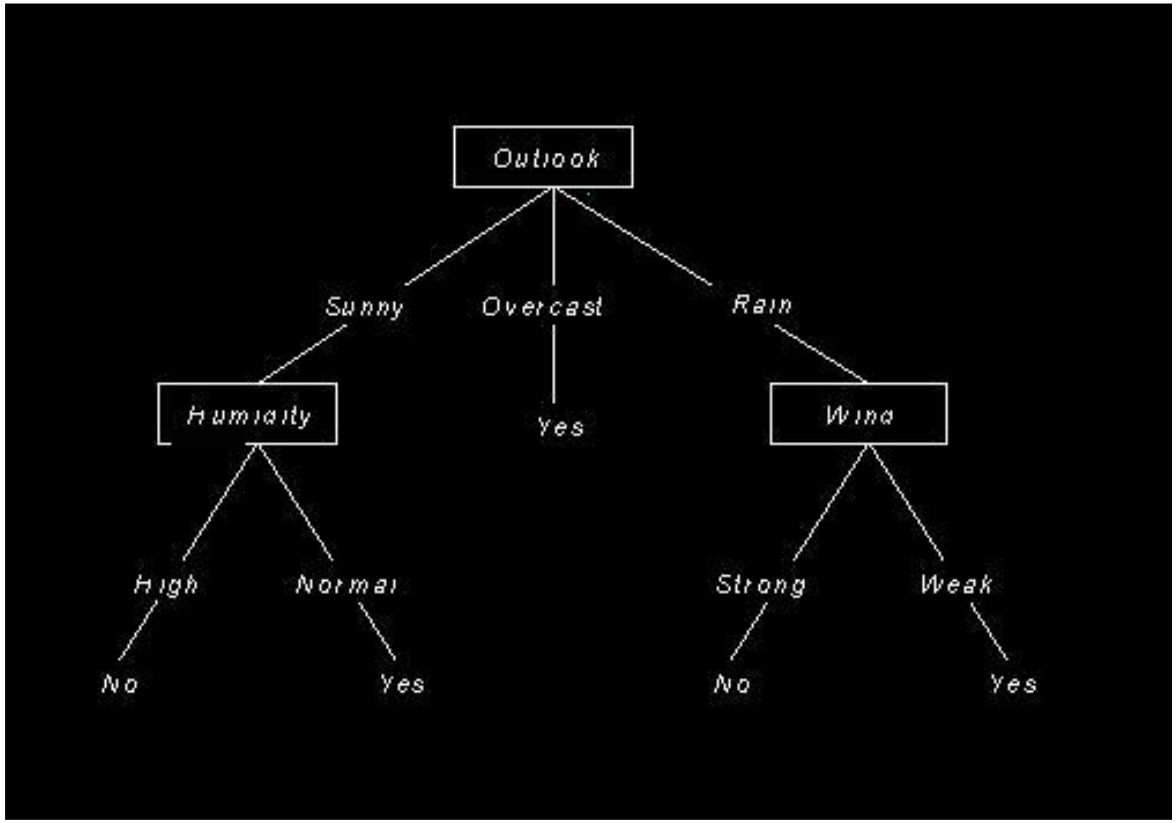
Medindo Desempenho

- **Processo de avaliação:**
  - **(1)** Divide-se o conjunto total de exemplos conhecidos em dois conjuntos:
    - Conjunto de Treinamento.
    - Conjunto de Teste.
  - **(2)** Gera-se uma hipótese h (árvore de decisão) com base no Conjunto de Treinamento.
  - **(3)** Para cada exemplo do Conjunto de Teste, classifica-se o exemplo utilizando a árvore de decisão criada a partir do conjunto de treinamento.
  - **(4)** Verifica-se a quantidade de exemplos de teste classificados corretamente e calcula-se a porcentagem de acertos.
  - **(5)** Escolhe-se aleatoriamente um novo conjunto de exemplos de treinamento (normalmente com um numero maior de exemplos) e repete-se novamente o processo.

Medindo Desempenho



Convertendo uma árvore em regras



Convertendo uma árvore em regras

- IF (Outlook = Sunny) ^ (Humidity = High)
  THEN PlayTennis = No

- IF (Outlook = Sunny) ^ (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.

Algoritmos para Árvores de Decisão

- ID3 (Iterative Dichotomiser 3)
  - Proposto por Ross Quinlan em 1986 no artigo “Induction of Decision Trees”
  - Apenas classes discretas
- C4.5
  - Baseado no ID3
  - Proposto por Ross Quinlan em 1993 no artigo “C4.5: Programs for Machine Learning”
  - Gera regras para classes discretas ou contínuas
- C4.5 foi eleito o algoritmo mais popular pela lista dos Top 10 Algoritmos p/ Data Mining pre-eminentes, publicado pela Springer LNCS em 2008
- No Weka, a implementação do C4.5 chama-se J48 pruned tree.

Leitura Complementar

- Russell, S. and Norvig, P. **Artificial Intelligence: a Modern Approach**, 3nd Edition, Prentice-Hall, 2009.

- **Capítulo 18: Learning from Observations**