ALGORITMOS DE INDUÇÃO DE ÁRVORES
COMO CALCULAR O GANHO?

Cristiane Neri Nobre

Árvore de Decisão – Como gerar?

Como avaliar **matematicamente** o **ganho** de cada atributo?

Árvore de Decisão – Como gerar?

Na maioria dos indutores de árvores de decisão, as funções de divisão discreta são univariadas, isto é, um nó interno é dividido de acordo com o valor de um único atributo.

Consequentemente, o indutor procura o melhor atributo sobre o qual realizar a divisão.

Árvore de Decisão – Como gerar?

Entropia é o cálculo do ganho de informação baseado em uma medida utilizada na teoria da informação.

Caracteriza a (im)pureza dos dados: em um conjunto de dados, é uma medida da falta de homogeneidade dos dados de entrada em relação a sua classificação.

Dado um conjunto de entrada (S) que pode ter *c* classes distintas, a entropia de *S* será dada por

$$Entropia(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Árvore de Decisão – Como gerar?

Assim, a entropia é uma medida da **aleatoridade** de uma variável

A física usa o termo entropia para descrever a quantidade de desordem associada a um sistema. Na teoria da informação, este termo tem uma significado semelhante, -- ele mede o grau de desordem de um conjunto de dados.

Escolha de testes de atributos

• O esquema usado na aprendizagem de árvores de decisão para selecionar atributos é projetado para **minimizar** a **profundidade** da árvore final.

• A ideia é escolher o **atributo** que **vá o mais longe** possível na tentativa de fornecer uma classificação exata dos exemplos.

• Um atributo perfeito divide os exemplos em conjuntos que são todos positivos ou todos negativos.

- *Clientes* – bastante bom
- *Tipo* – realmente inútil

Escolha de testes de atributos

• Assim, tudo o que precisamos é de uma medida formal de “bastante bom” e “realmente inútil”

• A função ESCOLHER-ATRIBUTO deverá ter seu valor **máximo** quando o **atributo for perfeito**, e seu valor **mínimo** quando o atributo for **absolutamente inútil**.

• Uma medida apropriada é a quantidade esperada de informações fornecidas pelo atributo, que é calculada através de uma expressão matemática

Escolha de testes de atributos

• Para se entender a **noção de informações**, pode-se pensar como a resposta a uma pergunta. Assim, a quantidade de informações contidas na resposta depende do conhecimento anterior do indivíduo.

• Quanto menos se sabe, mais informações são fornecidas.

• A **teoria da informação** mede o conteúdo de informação em bits.

Escolha de testes de atributos

• Um **bit de informação** é suficiente para responder a uma pergunta do tipo sim/não sobre a qual não se tem nenhuma ideia. Por exemplo se lançarmos uma moeda imparcial qual a quantidade de informação necessária?

Em geral, se cada resposta possível v_i têm probabilidade $P(v_i)$, então o conteúdo de informação "I" da resposta real é dado por:

$$I(P(V_1),\ldots,P(v_n)) = \sum_{i=1}^{n} - P(v_i)\log_2 P(v_i)$$

Escolha de testes de atributos

- No caso do lançamento de uma moeda imparcial, temos:

$$I(\frac{1}{2},\frac{1}{2}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

$$I(\frac{1}{2},\frac{1}{2}) = -\frac{1}{2}(-1) - \frac{1}{2}(-1)$$

$$I(\frac{1}{2},\frac{1}{2}) = \frac{1}{2} + \frac{1}{2} = 1\,\text{bit}$$

Ou seja**, um bit de informação** é suficiente para responder a uma pergunta **sim/não** sobre a qual não se tem nenhuma ideia, como o lançamento de uma moeda imparcial.

Escolha de testes de atributos

- E se a moeda for adulterada (viciada) para dar 99% de cara?

$$I(\frac{1}{100},\frac{99}{100}) = -\frac{1}{100}\log_2\frac{1}{100} - \frac{99}{100}\log_2\frac{99}{100}$$

$$I(\frac{1}{100},\frac{99}{100}) = -\frac{1}{100}(-0{,}0145) - \frac{99}{100}(-6{,}64386)$$

$$I(\frac{1}{100},\frac{99}{100}) = 0{,}014355 + 0{,}066439 = 0{,}080793$$

Escolha de testes de atributos

- Ou seja, como a **probabilidade de caras tente a 1**, a **informação da resposta** tende a 0.

- Para a aprendizagem em árvores de decisão, a pergunta que precisa ser respondida é: para um dado exemplo, qual é a **classificação correta**?

Escolha de testes de atributos

• Assim, uma estimativa das probabilidades das respostas possíveis antes de quaisquer atributos serem testados é dada pelas proporções de exemplos positivos e negativos no conjunto de treinamento.

Vamos supor que o conjunto de treinamento contenha *p* exemplos positivos e *n* exemplos negativos. Então uma estimativa das informações contidas em uma resposta correta é:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Escolha de testes de atributos

• Um teste em um único atributo A normalmente nos fornecerá algumas informações.

• Podemos medir exatamente quantas informações ainda precisaremos depois do teste do atributo.

• Qualquer atributo A divide o conjunto de treinamento E em subconjuntos E_1, …, E_v de acordo com seus valores para A, onde A pode ter *v* valores distintos.

Escolha de testes de atributos

• Cada subconjunto E_i tem P_i exemplos positivos e n_i exemplos negativos

• Assim, se seguirmos ao longo dessa ramificação, precisaremos de

$$I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

bits de informação para responder à pergunta.

Escolha de testes de atributos

• Um exemplo escolhido ao acaso a partir do conjunto de treinamento tem o i-ésimo valor para o atributo com probabilidade $(p_i+n_i)/(p+n)$ e assim, em média, depois de testar o atributo A, precisaremos de:

$$\text{Entropia(atributo)} = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

bits de informação para classificar o exemplo.

Escolha de testes de atributos

• O **ganho de informação** a partir do teste de atributo é a diferença entre o requisito de informação original e o novo requisito:

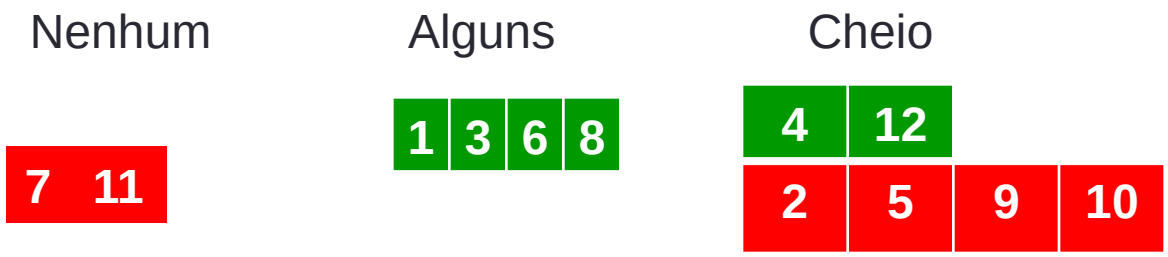
$$ganho(atributo) = Entropia(classe) - Entropia(atributo)$$

• A heurística usada na função ESCOLHER-ATRIBUTO é simplesmente escolher o atributo com o **maior ganho**.

Escolha de testes de atributos

Voltando aos atributos considerados no problema citado, qual o ganho do atributo **cliente**?

Escolha de testes de atributos



$$ganho(clientes) = 1 - \left[\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right) \right] \approx 0{,}541$$

Escolha de testes de atributos

E do atributo **Tipo**?

Escolha de testes de atributos

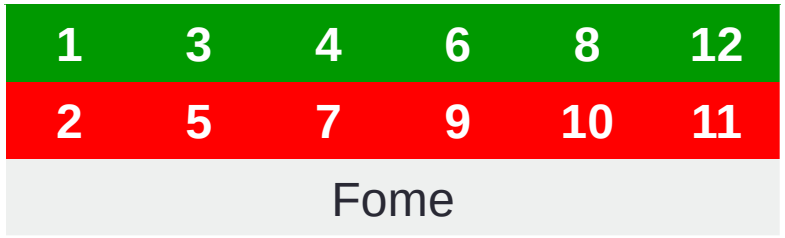
Francês	Italiano	Tailandês	Hambúrger
1	6	4 8	3 12
5	10	2 11	7 9

$$ganho(tipo) = 1 - \left[\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) \right] = 0$$

Escolha de testes de atributos

E do atributo **Fome**?

Escolha de testes de atributos



Exemplo	Alternativo	Bar	Sex/Sab	fome	Cliente	Preço	Chuva	Res	Tipo	Tempo	Conclusão (Vai esperar)?
X1	Sim	Não	Não	Sim	Alguns	RRR	Não	Sim	Francês	0-10	Sim
x2	Sim	Não	Não	Sim	Cheio	R	Não	Não	Tailandês	30-60	Não
x3	Não	Sim	Não	Não	Alguns	R	Não	Não	Hamburger	0-10	Sim
x4	Sim	Não	Sim	Sim	Cheio	R	Sim	Não	Tailandês	10-30	Sim
X5	Sim	Não	Sim	Não	Cheio	RRR	Não	Sim	Francês	>60	Não
X6	Não	Sim	Não	Sim	Alguns	RR	Sim	Sim	Italiano	0-10	Sim
X7	Não	Sim	Não	Não	Nenhum	R	Sim	Não	Hamburger	0-10	Não
X8	Não	Não	Não	Sim	Alguns	RR	Sim	Sim	Tailandês	0-10	Sim
X9	Não	Sim	Sim	Não	Cheio	R	Sim	Não	Hamburger	>60	Não
X10	Sim	Sim	Sim	Sim	Cheio	RRR	Não	Sim	Italiano	10-30	Não
X11	Não	Não	Não	Não	Nenhum	R	Não	Não	Tailandês	0-10	Não
X12	Sim	Sim	Sim	Sim	Cheio	R	Não	Não	Hamburger	30-60	Sim

Escolha de testes de atributos

1	3	4	6	8	12
2	5	7	9	10	11

Fome

Sim

1	4	6	8	12
2	10			

Não

3			
5	7	9	11

$$ganho(fome) = 1 - \left[\frac{7}{12} I(\frac{5}{7}, \frac{2}{7}) + \frac{5}{12} I\left(\frac{1}{5}, \frac{4}{5}\right) \right] \approx 0{,}196$$

Escolha de testes de atributos

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

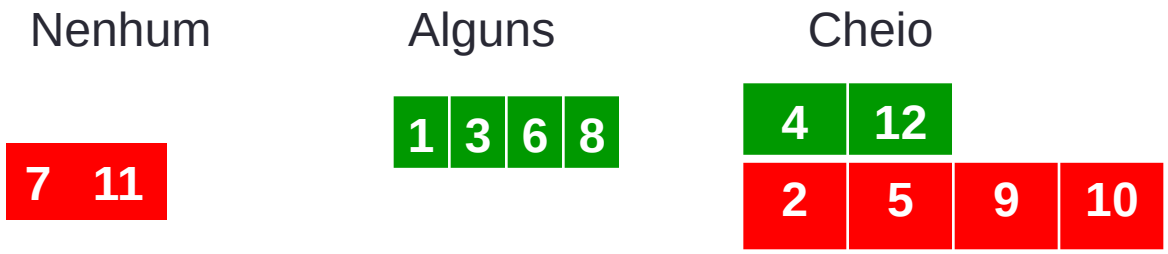
Escolha de testes de atributos

Calcule o ganho para todos os atributos desta base de dados

Quem será a raiz da árvore?

Como será feita a seleção do próximo atributo?

Escolha de testes de atributos



Ele calculará os valores:
Ganho(cheio, alt)
Ganho(cheio,bar)
Ganho(cheio,chu)…

E selecionará o melhor…

Escolha de testes de atributos

Gere a árvore para essa base de dados e confira os resultados no WEKA

Faça o mesmo para a base “Jogar tênis ou não”

Como avaliar a qualidade do modelo?

Como avaliar a **qualidade** do modelo obtido?

O que significam as **métricas** apresentadas?

Resumindo: como funciona o algoritmo?

A ideia base do algoritmo é:

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.
    2. Senão repetir os passos 1 a 4

Árvore de Decisão – Algoritmo

O Algoritmo ID3

- Foi criado por Ross Quinlan, na universidade de Sydnei, Austrália em 1979
- O ID3 usa entropia e ganho de informação para construir a árvore de decisão
- É o algoritmo pioneiro em indução de árvore de decisão.
- É recursivo, baseado em busca gulosa, onde procura um conjunto de atributos que melhor dividem os exemplos (amostras), gerando sub-árvores.

Árvore de Decisão – Algoritmo

ID3(*Exemplos*, *Atributo-objetivo*, *Atributos*)

// **ID3** retorna uma árvore de decisão que classifica corretamente os *Exemplos* determinados

// *Exemplos* são os exemplos de treinamento.

// *Atributo-objetivo* é o atributo cujo valor deve ser predito pela árvore.

// *Atributos* são uma lista de outros atributos que podem ser testados pela árvore de decisão.

Início

Crie um nodo *Raiz* para a árvore

Se todos os *Exemplos* são positivos

Então retorna a *Raiz* da árvore com o rótulo = **sim**

Se todos os *Exemplos* são negativos

Então retorna a *Raiz* da árvore com o rótulo = **não**

Se *Atributos* for vazio

Então retorna a *Raiz* da árvore com o rótulo = valor mais comum do *Atributo-objetivo* em *Exemplos*

Senão

A ← um atributo de *Atributos* que melhor classifica *Exemplos* (atributo de decisão)

Raiz ← **A** (rótulo = atributo de decisão **A**)

Para cada possível valor v_i de **A** faça

Acrescenta um novo arco abaixo da *Raiz*, correspondendo à resposta **A** = v_i

Seja $Exemplos_{vi}$ o subconjunto de *Exemplos* que têm valor v_i para **A**

Se $Exemplos_{vi}$ for vazio

Então acrescenta na extremidade do arco um nodo folha

com rótulo = valor mais comum do *Atributo-objetivo* em *Exemplos*

Senão acrescenta na extremidade do arco a sub árvore

ID3($Exemplos_{vi}$, *Atributo-objetivo*, *Atributos* – {**A**})

Retorna *Raiz* (aponta para a árvore)

Fim

Árvore de Decisão – Algoritmo

Limitações do Algoritmo ID3

- A principal limitação do ID3 é que ele só lida com atributos discretos, não sendo possível apresentar a ele um conjunto de dados com atributos contínuos. Neste caso, os atributos devem ser discretizados.

- O ID3 também não apresenta nenhuma forma de tratar valores desconhecidos, ou seja, todos os exemplos do conjunto de treinamento deve ter valores conhecidos para todos os seus atributos

Árvore de Decisão – Algoritmo

Limitações do Algoritmo ID3

- O algoritmo também não lida com nenhum mecanismo pós-poda, o que poderia amenizar em árvores mais complexas

Árvore de Decisão – Algoritmo

C4.5 é uma extensão do algoritmo ID3, ambos desenvolvido por Ross Quinlan.
Ele apresenta uma série de melhorias do algoritmo ID3.

Algumas destas melhorias são:

- Lidar com atributos contínuos e discretos. No intuito de lidar com atributos contínuos, o algoritmo C4.5 cria um valor limiar e então particiona o conjunto de dados em dois subconjuntos dos quais um contém valores de atributos maiores do que aquele valor limiar e o outro conjunto contém valores menores ou iguais aquele valor limiar.

- Lidar com dados de treinamento com atributos incompletos. O algoritmo C4.5 permite que atributos sejam rotulados como *?* para casos onde os valores não estejam presentes. Valores de atributos que não estejam presentes não são utilizados em cálculos de entropia ou ganho de informação.

- Poda de árvores após a criação. O algoritmo C4.5 retrocede pela árvore quando esta é criada e tenta remover ramificações que não ajudam no processo de decisão e substitui estes ramos por nós folha.

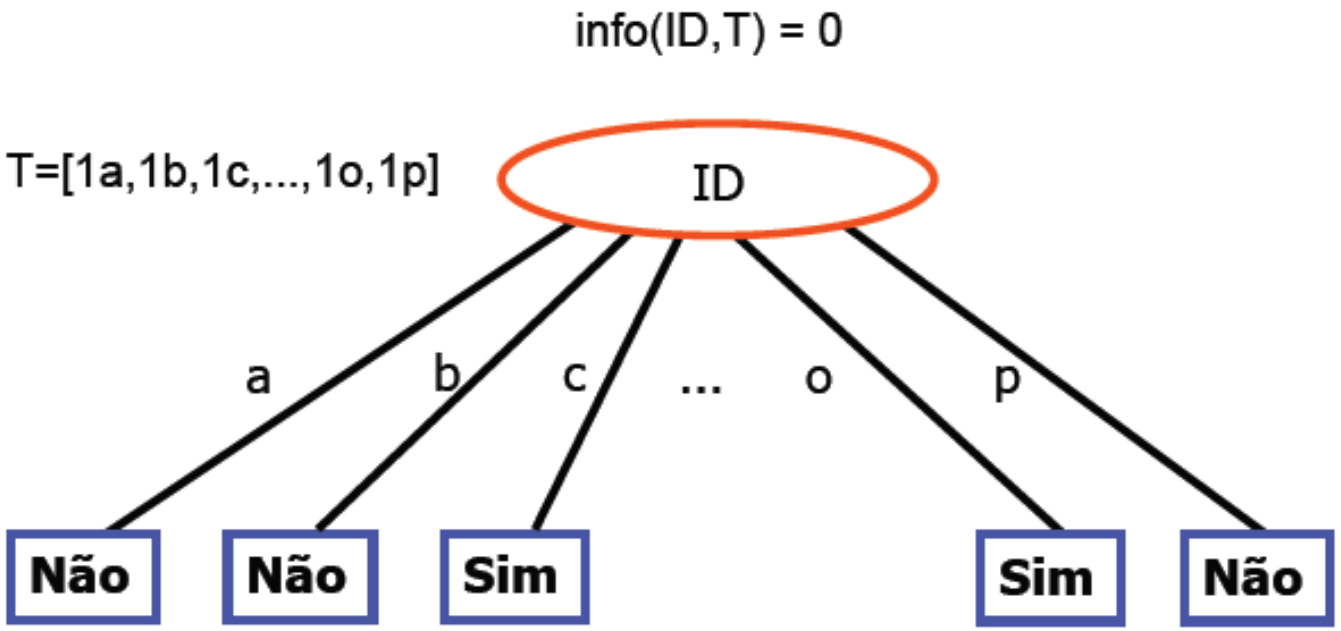
- Implementa o 'gain ratio' ao invés do 'ganho de informação' tradicional

Razão de ganho (gain ratio)

Uma vez que cada código ID é único, particionando o conjunto de treinamento nos valores deste atributo levará a um grande número de subconjuntos, cada um contendo somente um caso

Como todos os subconjuntos (de 1 elemento) necessariamente contêm exemplos de uma mesma classe, **info(ID,T)=0**, assim o ganho de informação deste atributo será máximo

Razão de ganho (gain ratio)



Razão de ganho (gain ratio)

Para solucionar esta situação, em analogia à definição de info(T), vamos definir a informação potencial gerada pela partição de T em r subconjuntos

$$\text{split-info}(X,T) = -\sum_{i=1}^{r} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}$$

A razão de ganho é definida como:

$$\text{gain-ratio}(X,T) = \frac{\text{gain}(X,T)}{\text{split-info}(X,T)}$$

A **razão de ganho** expressa a proporção de informação gerada pela partição que é útil, ou seja, que aparenta ser útil para a classificação

Razão de ganho (gain ratio)

Usando o exemplo anterior para o atributo Clientes que produz três subconjuntos com 2, 4 e 6 exemplos, respectivamente

$$ganho(clientes) = 1 - \left[\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right)\right] \approx 0{,}541\, bits$$

$$split-\inf o(Clientes, T) = -\frac{2}{12}\log_2 \frac{2}{12} - \frac{4}{12}\log_2 \frac{4}{12} - \frac{6}{12}\log_2 \frac{6}{12}$$

$$split-\inf o(Clientes, T) = -\frac{2}{12}(-2.58496) - \frac{4}{12}(-1.58496) - \frac{6}{12}(-1) = 1.459$$

Razão de ganho (gain ratio)

Para este teste, cujo ganho é gain(Clientes,T) =0.541 (mesmo valor anterior), a razão de ganho é:

gain-ratio(Clientes,T) = $\frac{0,541}{1,459} = 0,37$

Assim, a **razão de ganho** expressa a proporção de informação gerada pela partição que é útil, ou seja, que aparenta ser útil para a classificação

Como o C4.5 lida com valores numéricos?

Um teste em um atributo numérico produz uma partição binária do conjunto de exemplos:

- Exemplos onde valor_do_atributo <= ponto_referência
- Exemplos onde valor_do_atributo > ponto_referência

Como o C4.5 lida com valores numéricos?

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 treino pode ser utilizado como possível ponto de referência.
  - ✓ É usual considerar o valor médio entre dois valores diferentes e consecutivos.
  - ✓ Soma-se o valor menor e o valor maior e divide-se pelo número de classes
- 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.