Fundamentos de Árvores de Decisão

Parte I

1. Árvores de Decisão

- De um ponto de vista formal, uma **árvore** é um grafo não-direcionado no qual dois vértices quaisquer se conectam por um único caminho (um grafo acíclico não-direcionado) [Wikipedia, 2019]. Trata-se de uma estrutura de dados de grande importância para a computação em geral e para as áreas de aprendizado de máquina, tomada de decisão e teoria de jogos em particular.
- A árvore possui um nó raiz, do qual parte o processo de decisão / partição. Nesse processo, valores distintos de atributos geram arestas (ramificação), e, quando se chega a um nó folha, ocorre uma atribuição de classe.

- Na Fig. 1, temos um exemplo bastante simples. Do *nó raiz*, saem duas arestas que levam aos nós I1 e I2 (nós *intermediários*, como veremos melhor em seguida). Essa ramificação, via de regra, vincula-se a uma decisão como a análise de um valor (e.g. “Maior que 5? Sim ou Não?”) ou a tomada de um percurso. De I1 saem dois ramos que levam aos nós F1 e F2. Esses nós são chamados de *folhas*, pois são nós terminais, dos quais não saem mais ramificações. De I2 também é possível chegar a dois nós-folha, F3 e F4.

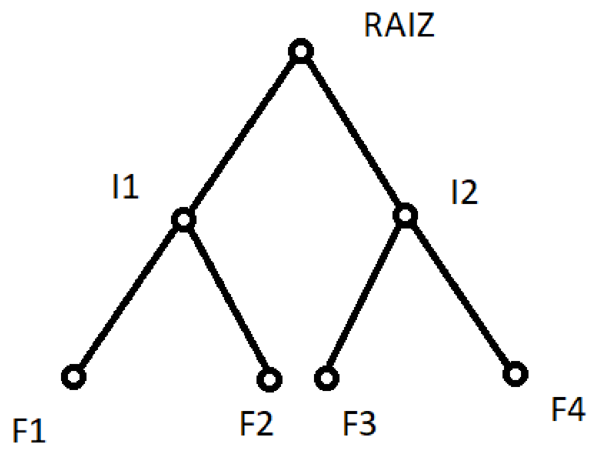

Figura 1 – Exemplo de Árvore

- Na Fig. 1.b, temos a imagem de uma árvore construída a partir de um conjunto de dados denominado "Titanic".

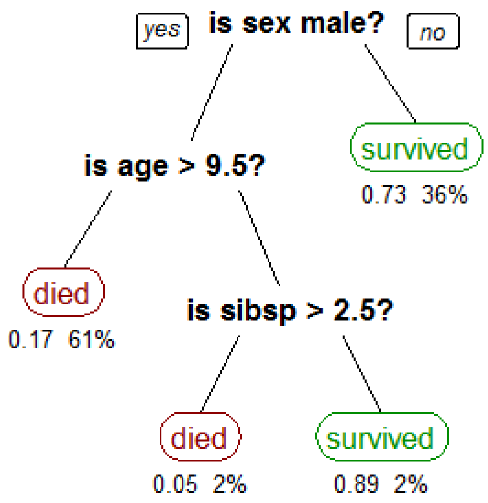

Figura 1.b - Exemplo de Árvore de Decisão (de [Wikipedia, 2019]).

- Cada atributo, no caso, leva a uma resposta binária (sim / não), e, para cada nó-folha, atinge-se uma decisão sobre morte e sobrevivência.

- O uso da árvore para classificar padrões é relativamente direto, mas é preciso dar uma resposta a uma questão crucial: como induzir uma árvore de decisão a partir de dados? Para que possamos dar uma resposta válida a essa questão, seguiremos aproximadamente o curso do artigo seminal de J. R. Quinlan [Quinlan, 1986]. Partiremos, desse modo, de um exemplo dado por ele.

1.1. Exemplo de Árvore de Decisão

- Consideremos um conjunto de dados da forma $(\mathbf{x}_i, d_i)$, onde $\mathbf{x}_i$ é um vetor de atributos e d_i é um rótulo. Nesse conjunto, cada entrada diz respeito à condição meteorológica de um dia. Os atributos são:
  - ***Tempo:*** *{ensolarado, nublado, chuvoso}*
  - ***Temperatura:*** *{dia frio, dia agradável, dia quente}*
  - ***Umidade:*** *{alta, normal}*
  - ***Vento:*** *{presente, ausente}*
- Os rótulos, por sua vez, são apenas "positivo" (P) e "negativo" (N), denotando um problema genérico com duas classes. Um exemplo de manhã poderia ser descrito da seguinte forma: {nublado, dia frio, normal, ausente}.
- O conjunto de treinamento é a base para definir a árvore. Um conjunto que contenha inconsistências, como dois padrões com os mesmos atributos e classes

diferentes, precisará ser reconsiderado (os atributos podem não ser suficientes, por exemplo).

- Um conjunto de treinamento possível é dado na Fig. 2.

No.	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

Figura 2. Possível Conjunto de Treinamento (de [Quinlan, 1986]).

- Apenas para mostrar o objetivo de projeto, apresentamos, na Fig. 3, uma árvore que classifica corretamente os exemplos do conjunto de dados.

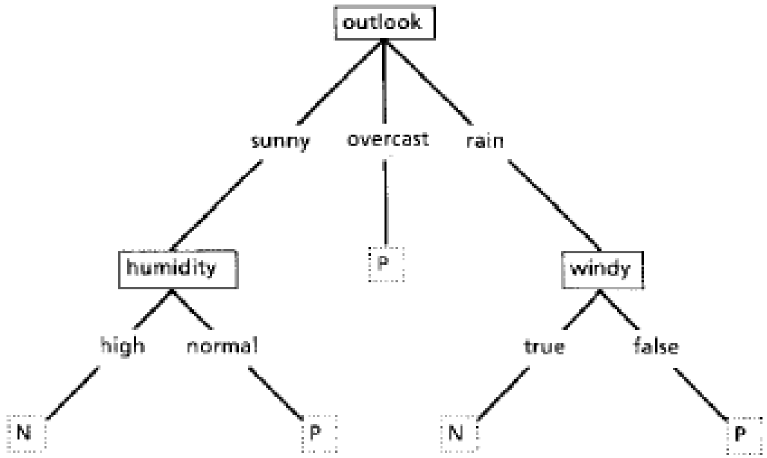

Figura 3. Exemplo de Árvore de Decisão.

- No projeto de uma árvore, é necessário ter em conta o princípio que norteia o aprendizado de máquina em geral: uma estrutura demasiadamente complexa pode significar que o conjunto de treinamento foi "aprendido" de maneira artificial, ou seja, que houve sobreajuste. Portanto, o princípio da *navalha de Ockham* permeia o projeto de árvores de decisão.

- Para ilustrar esse ponto, apresentamos, na Fig. 4, uma árvore que também explica os dados, mas que é significativamente mais rebuscada. Essa árvore não seria desejável.

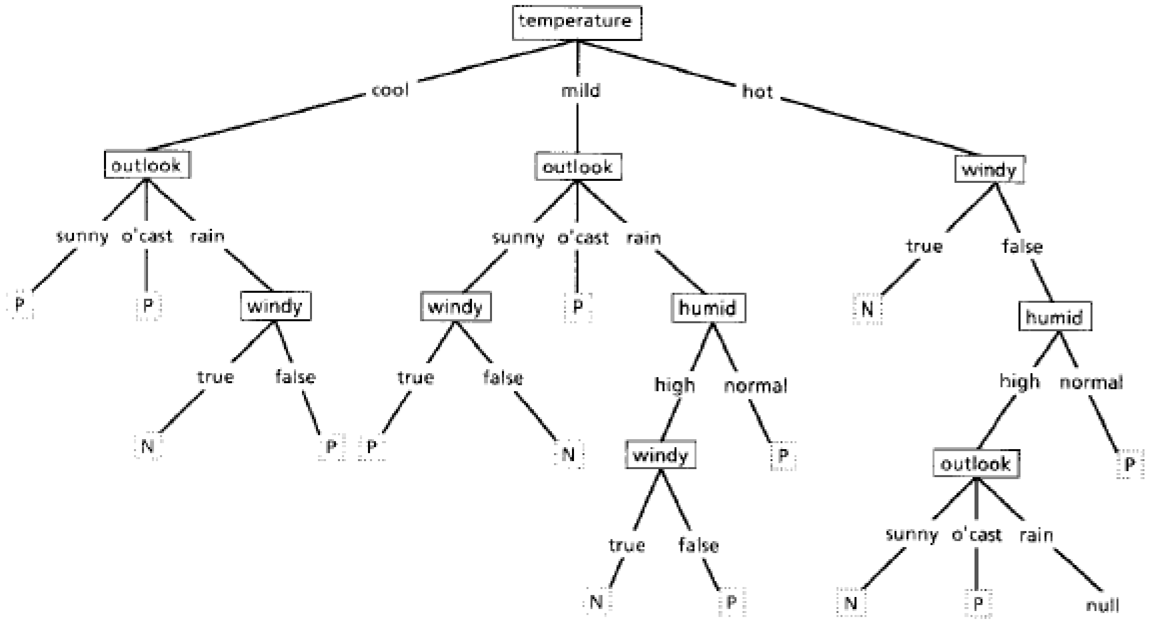

Figura 4. Árvore Complexa (de [Quinlan, 1986]).

1.2. O Processo de Indução (Exemplo do Método ID3)

- Uma primeira abordagem poderia ser construir, de maneira exaustiva, todas as árvores capazes de resolver determinado problema e selecionar a mais simples. Essa abordagem, no entanto, pode ser demasiadamente custosa. O método ID3 (*Iterative Dichotomiser* 3), que discutiremos a seguir, é uma abordagem que não garante a obtenção da menor árvore, mas busca obter árvores apropriadas num período de tempo relativamente curto.
- No método, escolhe-se aleatoriamente um subconjunto dos dados de treinamento (janela) e se constrói uma árvore que o representa. São, então, apresentados os demais padrões do conjunto de treinamento: caso eles também sejam adequadamente classificados, a árvore estará pronta; caso contrário, uma seleção dos dados classificados incorretamente é adicionada à janela e se repete o processo de construção da árvore.

- Vejamos como a árvore é construída a partir de uma coleção de exemplos. Consideremos um teste T que seja feito sobre determinado atributo, com possíveis resultados O_1, $O_2, \ldots, O_w$. Cada padrão no conjunto C terá esses resultados para o teste T, de modo que surge uma partição $\{\mathsf{C}_1, \ldots, \mathsf{C}_w\}$, como mostra a Fig. 5.

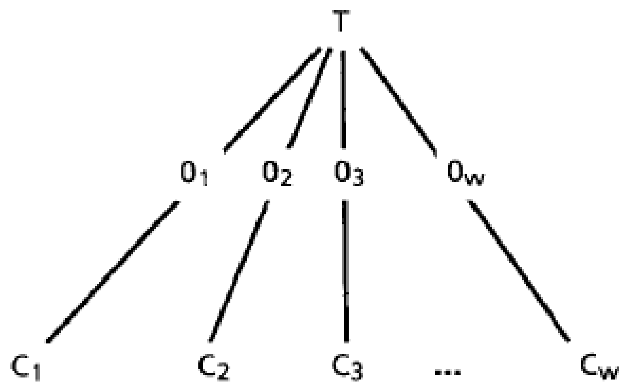

Figura 5. Partição (de [Quinlan, 1986]).

Dois pontos devem ser ressaltados:

- Se cada subconjunto C_i for associado a uma árvore de decisão, ter-se-á uma árvore mais ampla para todos os padrões.
- Se dois ou mais C_i's são não-vazios, cada C_i será menor que C.

- Como se seleciona o atributo a gerar a partição? A metodologia do ID3 é baseada na teoria da informação. Duas hipóteses são fundamentais (o valor p é o número de amostras da classe P e o valor n é o número de amostras da classe N):
  - Qualquer árvore de decisão correta para C classificará objetos na mesma proporção de ocorrência das classes no conjunto de dados. Assim, a probabilidade de um dado ser da classe P é $p/(p+n)$ e a de um dado ser da classe N é $n/(p+n)$.
  - Assim, a árvore de decisão pode ser vista como uma fonte binária de informação com entropia igual a:

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

- Consideremos um atributo A com valores $\{A_1, \dots, A_v\}$ a ser usado junto ao nó raiz. Esse atributo particionará os dados em conjuntos $\{C_1, \dots, C_v\}$. Consideremos que

C_i contenha p_i objetos da classe P e n_i objetos da classe N. A informação média $E(A)$ associada ao atributo A como raiz é:

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- O ganho de informação obtido pela partição segundo o atributo A é:

$$\text{Ganho}(A) = I(p, n) - E(A)$$

- A ideia seria então maximizar esse ganho de informação e então usar o procedimento recursivamente para os subconjuntos $C_1, \dots, C_v$. Ou seja, escolhe-se o atributo que gera a primeira ramificação e, então, se repete o processo para construir as subárvores.

- Como exemplo, podemos avaliar os dados da Fig. 2. Há 14 padrões, 9 da classe P e 5 da classe N. A informação (entropia) é:

$$I(p, n) = -\frac{9}{14} log_2\left(\frac{9}{14}\right) - \frac{5}{14} log_2\left(\frac{5}{14}\right) = 0{,}940 \text{ bits}$$

- Consideremos agora o atributo "tempo" ("*outlook*"). Cinco padrões tem o valor "ensolarado", e, destes, dois são classe P e três da N. Assim, p_1 = 2, n_1 = 3 e $I(p_1, n_1)$ = 0,971 bits. Analogamente, p_2 = 4, n_2 = 0 e $I(p_2, n_2)$ = 0. Por fim, p_3 = 3, n_3 = 2 e $I(p_3, n_3)$ = 0,971. Portanto,

$$E(\text{'tempo'}) = \frac{5}{14}I(p_1, n_1) + \frac{4}{14}I(p_2, n_2) + \frac{5}{14}I(p_3, n_3) = 0{,}694 \text{ bits}$$

- O ganho do atributo é ganho('tempo') = 0,940 – E('tempo') = 0,246 bits. A análise dos atributos 'temperatura', 'umidade' e 'vento' leva a ganhos de, respectivamente, 0,029, 0,151 e 0,048 bits. Dessa forma, escolhe-se o atributo "tempo". Em seguida, dividem-se os padrões em subconjuntos de acordo com os valores do atributo escolhido e uma árvore de decisão é induzida para cada subconjunto de maneira idêntica. Isso leva exatamente à árvore da Fig. 3.

1.3. Uso do Índice de Gini

- Esse índice, proposto em 1912 pelo italiano Corrado Gini, é amplamente usado nas áreas de economia e ciência social no estudo da desigualdade de renda [Gini, 1921]. Ele também permite a análise dos atributos de uma árvore de decisão [Murthy, 1998].
- Nesse caso, costuma-se trabalhar com a seguinte forma:

$$Gini(p,n) = 1 - \left[\left(\frac{p}{p+n}\right)^2 + \left(\frac{n}{p+n}\right)^2\right]$$

Nesse caso, seria preciso avaliar o valor esperado para o atributo A

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} Gini(p_i, n_i)$$

e determinado o ganho de Gini associado:

$$\text{Ganho} = Gini(p, n) - E(A)$$

- O atributo com maior ganho é escolhido, como no caso em que se usa a informação.

1.4. Alguns Aspectos

- Caso haja dados ruidosos, ou seja, dados que não plenamente "consistentes", passa a ser necessária uma análise estatística mais ampla, incluindo, por exemplo, testes de hipóteses.
- Outro ponto importante é, se for o caso, buscar metodologias para lidar com atributos faltantes.

Referências bibliográficas

GINI, C., *Nota sobre o artigo "The Measurement of Inequality of Incomes", de H. Dalton*, The Economic Journal, Vol. 31, No. 121, pp. 124 – 126, 1921.

MURTHY, S. K., "Automatic Construction of Decision Trees from Data: a Multi-Disciplinary Survey", Data Mining and Knowledge Discovery, Vol. 2, No. 4, pp. 345 – 389, 1998.

QUINLAN, J. R., "Induction of Decision Trees", Machine Learning, Vol. 1, pp. 81 – 106, 1986.

WIKIPEDIA, Artigos Diversos, 2019.