ME731-Métodos em Análise Multivariada
Universidade Estudual de Campinas
Instituto de Matemática, Estatística e Computação Científica
Bacharelado em Estatítica

Caio Théodore Genovese Huss Oliveira - 247005

Árvores de Decisão para classificação de dados binários

Conteúdo

1 Introdução

Árvores de decisão são preditores $h : X \to K$ que atribuem uma etiqueta $k \in K$ a cada observação x de nossos dados através de uma série de decisões. Neste trabalho são analisadas respostas binárias, ou seja, $K = \{0\,, 1\}$. A predição de uma resposta binária é muito útil em várias áreas, como fraude, crédito e diagnóstico de doenças, entre outras.

Árvores de decisão também podem ser utilizadas para respostas com várias categorias ou contínuas, no caso de árvores de regressão.

Este trabalho almeja descrever as árvores de decisão em seus aspectos teóricos, com foco no caso binário, detalhar as vantagens e desvantagens do modelo, e mostrar um exemplo de aplicação.

2 Aspectos Teóricos

A árvore atribui uma etiqueta k à observação x_i fazendo a mesma "percorrer"os nós da árvore, começando pelo nó *raiz* e terminando em um dos nós *folha*, exemplificados na figura 1.

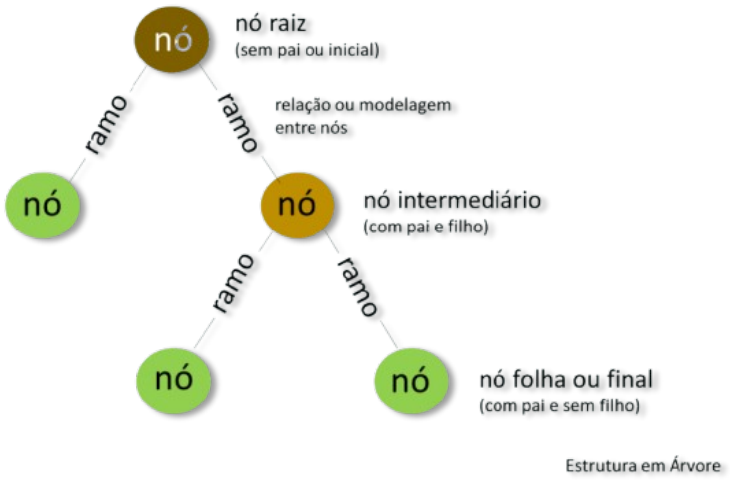

Figura 1: Elementos de uma árvore de decisão. Fonte: Blog Colaborae, https://colaborae.com.br/blog/2023/07/19/arvore-de-decisao/

Cada nó *raiz* ou *intermediário* faz uma divisão nos dados, com o objetivo de separá-los em grupos homogêneos com relação à variável resposta, o que frequentemente não é possível, aceitando-se grupos com proporções altas de uma das etiquetas. Nós *folha* são a "decisão" tomada, ou seja, a previsão para nossa observação.

Resta a definir como essas divisões são feitas, e o que é aceitável como uma "proporção alta".

2.1 Regras de Divisão

Todo nó *raiz* ou *intermediário* utiliza uma regra de divisão para separar as observações.

Uma regra popular, e a que será abordada neste trabalho, é a de escolher um valor θ para uma variável j, e dividir as observações com uma função indicadora $\mathbb{1}_{(x_{ij}>\theta)}(\overline{x}_i)$, onde $\overline{x}_i$ é o vetor das d variáveis explicativas da observação i.

$$\mathbb{1}_{(X_{ij}>\theta)}(\widetilde{X}_i) = 1 \rightarrow \widetilde{X}_i \; passa \; ao \; no \; direito$$

$$\mathbb{1}_{(X_{ij}>\theta)}(\widetilde{X}_i) = 0 \rightarrow \widetilde{X}_i \; passa \; ao \; no \; esquerdo$$

Uma *função de ganho* é definida para escolher o valor ótimo de θ. A função mais simples é de erro de treino. Definindo a função $C(a) = min\{a, 1-a\}, a \in [0,1]$, o erro de treino antes de se fazer quaisquer divisões é $C(\mathbb{P}[k=1])$. Temos então, para o conjunto de dados S e a variável j:

$$Ganho(S, j, \theta) =$$

$$C(\mathbb{P}[k=1]) - (\mathbb{P}[\widetilde{X}^*_{j,\theta} = 0]C(\mathbb{P}[k=1|\overline{x}^*_{j,\theta} = 0] \; + \; \mathbb{P}[\widetilde{X}^*_{j,\theta} = 1]C(\mathbb{P}[k=1|\overline{x}^*_{j,\theta} = 1]))$$

onde $\widetilde{X}^*_{j,\theta} = \mathbb{1}_{(X_{ij}>\theta)}(\widetilde{X}_i)$, $\overline{x}^*_{j,\theta} = \mathbb{1}_{(x_{ij}>\theta)}(\overline{x}_i)$. queremos portanto o θ que resulte na maior diminuição do erro de treino, ou seja, o maior valor da função de ganho. As divisões seguintes seguem a mesma lógica.

Outras funções de ganho seguem a mesma lógia, modificando apenas a função $C(a)$:

- **Ganho de informação:** $C(a) = -a\,log(a) - (1-a)log(a)$
- **Índice de Gini:** $C(a) = 2a(1-a)$

2.2 Estrutura da árvore

Inicialmente, construímos a árvore pelo algoritmo **ID3**[1][2], recursivo:

- Para o conjunto de dados S de n obs., conjunto de variáveis explicativas A, e etiquetas $k_i \in \{0, 1\}$, ID3$\{S, A\}$:

1. Caso $k_i = k_j = h$, $\forall i, j = 1, ..., n$ **retorne** uma *folha* com o valor h, $h \in \{0, 1\}$.
2. Caso não existam variáveis explicativas, **retorne** uma folha com o valor da maioria das etiquetas k_i.
3. Caso contrário,
   (a) Defina $j = argmax_{i \in A} max_{\theta} Ganho(S, i, \theta)$.
   (b) Se a divisão gerada por j, θ retorna etiquetas iguais, **retorne** uma folha com o valor das etiquetas.
   (c) Caso contrário,
   - $T_1 = ID3\{\{(\overline{x}, y) \in S : \overline{x}^*_{j,\theta} = 0\}, A \backslash \{j\}\}$
   - $T_2 = ID3\{\{(\overline{x}, y) \in S : \overline{x}^*_{j,\theta} = 1\}, A \backslash \{j\}\}$
   - **Retorne** a árvore:

$$\overline{x}_j > \theta \rightarrow T_2, \quad \overline{x}_j \leq \theta \rightarrow T_1$$

Esta árvore ainda precisa ser reduzida, afim de evitar overfitting.

Uma das técnicas de redução comum é pegar uma árvore gerada pelo algoritmo ID3, e, começando do último nó e passando por todos, definir a árvore T' como a que minimiza uma função $f(T, m)$, entre as opções com relação ao nó j:

- Substituir o nó j por uma folha de valor $h \in \{0, 1\}$.
- Substituir o nó j pela árvore à esquerda ou à direita do nó ("remover" a divisão que gerou o nó j).
- Não mudar nada.

Onde n é o número de observações.

Substitui-se, então, a árvore T pela nova árvore T'. Neste trabalho a função $f(T, m)$ não será discutida.

[1] Iterative Dichotomizer 3

3 Aplicação

3.1 Dados e objetivo

Todos os códigos foram feitos na linguagem R, e os pacotes utilizados na aplicação estão mencionados em [1]. Os dados utilizados estão disponíveis neste link, obtidos do Kaggle.

O banco contém 568 observações de diagnósticos de câncer de mama, contendo informações visuais sobre o câncer e o tipo, maligno ou benigno. Mais detalhadamente, temos as variáveis:

- **Id:** Id único de cada paciente (descartada).
- **diagnosis:** Diagnóstico, "B" para benigno ou "M" para maligno.
- A média, o mínimo e o máximo das seguintes variáveis, todas numéricas e contínuas:
  - **radius**: raio do tumor.
  - **texture**: textura do tumor.
  - **perimeter**: perímetro do tumor.
  - **area**: área do tumor.
  - **smoothness**: "suavidade" do tumor.
  - **compactness**: o quão o tumor é compacto.
  - **concavity**: concavidade do tumor.
  - **concave points**: número de regiões côncavas do tumor.
  - **symmetry**: simetria do tumor.
  - **fractal_dimmension**: "coastline approximation"
- uma coluna vazia (descartada).

Mais informações podem ser encontradas na página do Kaggle[3].

O objetivo, neste relatório, é modelar uma árvore de decisão para diagnosticar um paciente com câncer maligno ou benigno, com base nas outras 30 variáveis[2].

Os dados serão divididos em conjunto de treino (40%, 227 obs.) e conjunto de teste (60%, 341 obs.), escolhidos aleatóriamente. O primeiro será utilizado para construir a árvore, e o segundo para testar sua predição, verificar se não houve *overfitting*, e outras análises necessárias.

[2]Na prática, bem menos serão necessárias

3.2 Treino de modelo

Com a função `rpart()`, construiu-se a árvore na figura 2

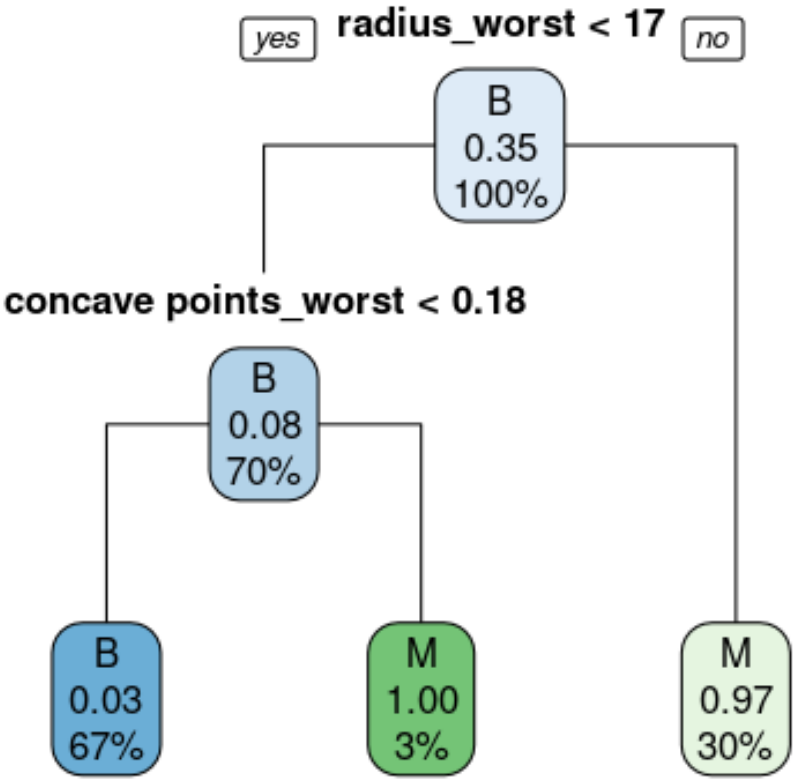

Figura 2: Árvore de decisão gerada com o conjunto treino.
Os nós contém, respectivamente: O valor predominante no nó, a proporção de obs. com câncer Maligno, e a porcentagem de obs. no nó, com relação ao banco total.

Observa-se que apenas duas variáveis foram relevantes para o modelo: O pior (máximo) raio do tumor e o pior (máximo) número de regiões côncavas em um tumor, resultando em 3 grupos cou pouca variação.

A árvore contém 5 nós, sendo 3 *folhas*, 1 nó *intermediário* e uma *raíz*. Sob este modelo, 67% do banco de treino é diagnosticado com câncer beningno, e o restante com câncer maligno. Para este banco, as previsões resultam em 5 falsos negativos (para a predição de câncer maligno) e 2 falsos positivos.

3.3 Previsão

Passando ao banco de teste, utiliza-se o modelo construído para predizer os diagnósticos do banco de teste, obtendo os seguintes resultados:

$$Taxa\ de\ acertos = 92,08211\% \quad Sensibilidade^* = 96,49123\% \quad Especificidade^* = 89,86784\%$$

*: com relação ao diagnóstico de câncer maligno.

As tabelas 1 e 2 possuem as contagens e proporções completas, respectivamente.

		Diagnóstico		
		B	M	Total
Previsão	B	204	23	227
	M	4	110	114
	Total	208	133	341

Tabela 1: Matriz de confusão para a árvore de decisão

		Diagnóstico		
		B	M	Total
Previsão	B	0.598	0.067	0.666
	M	0.012	0.323	0.334
	Total	0.610	0.390	1.000

Tabela 2: Proporções da matriz de confusão, com relação à amostra total

4 Vantagens e Desvantagens

Visto a aplicação com relação á uma variável binária, é interessante comparar a árvore de decisão com a regressão logística, outra técnica utilizada para a explicação da relação entre variáveis explicativas e uma variável binária, e para a predição de variáveis binárias.

Com tal comparação, e com alguns aspectos gerais da árvore de decisão, temos:

4.1 Vantagens

- Simplicidade de implementação, comunicação e decisão: As divisões da árvore de decisão são simples de serem entendidas, e mais fáceis de serem comunicadas do que parâmetros em um modelo de regressão. Uma pessoa com pouco conhecimento matemático consegue seguir a árvore e entender o resultado.

- Facilidade de visualização: A árvore proporciona um gráfico simples com cada decisão, facilitando a comunicação dos resultados obtidos.

4.2 Desvantagens

- Perda de informação: Ao trocar um coeficiente por uma simples divisão, perdemos informação sobre o relacionamento entre as variáveis, sendo menos interessante para fins de modelagem explicativa.
- Overfitting: A tendência dos algoritmos a construir árvores muito grandes resulta em overfitting, sendo necessário uma atividade de "correção" após a modelagem.
- Falta de rigor estatístico: Em minha busca, encontrei pouca informação sobre testes ou estatísticas para testes em cima de árvores de decisão, não encontrando meios de testar hipóteses (por exemplo, $H_0 : j$ é um nó intermediário, $H_A : j$ é uma folha).

Referências

[1] Pacotes do R utilizados: Pacman, Tidyverse, rpart, rpart.plot.

[2] Shai Shalev-Shwartz e Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms, cap. 18*. Cambridge University Press, 2014.

[3] https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.