Fundamentos Matemáticos dos Algoritmos de Machine Learning: O Algoritmo Naive Bayes

Fernando Lima Filho

18 de junho de 2025

Resumo

Este documento explora os fundamentos matemáticos do algoritmo Naive Bayes, um classificador probabilístico amplamente utilizado em Machine Learning. Abordamos o Teorema de Bayes, a suposição de independência condicional entre os atributos e demonstramos como o modelo calcula as probabilidades para realizar previsões.

1 Introdução ao Naive Bayes

O Naive Bayes é uma família de algoritmos de classificação baseados no Teorema de Bayes. Apesar de sua simplicidade e da suposição "ingênua"(daí o nome "naive") de que os atributos (features) são condicionalmente independentes, ele se mostra surpreendentemente eficaz em muitas aplicações do mundo real, como filtragem de spam, classificação de documentos e diagnóstico médico.

Sua principal vantagem reside na baixa exigência computacional, tanto para treinamento quanto para classificação, e na sua capacidade de lidar bem com um grande número de atributos.

2 O Fundamento: Teorema de Bayes

A base do Naive Bayes é o Teorema de Bayes, que descreve a probabilidade de um evento, com base em conhecimento prévio de condições que podem estar relacionadas a esse evento. A fórmula do teorema é:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Onde:

- $P(A|B)$ é a **probabilidade a posteriori**: a probabilidade do evento A ocorrer, dado que o evento B já ocorreu.
- $P(B|A)$ é a **verossimilhança (likelihood)**: a probabilidade do evento B ocorrer, dado que o evento A já ocorreu.

- $P(A)$ é a **probabilidade a priori**: a probabilidade do evento A ocorrer, independentemente de B.
- $P(B)$ é a **evidência (evidence)**: a probabilidade marginal do evento B ocorrer.

3 Aplicando o Teorema de Bayes à Classificação

No contexto de Machine Learning, queremos usar o Teorema de Bayes para encontrar a classe mais provável para um determinado conjunto de atributos. Podemos adaptar a fórmula da seguinte maneira:

$$P(\text{classe}|\text{atributos}) = \frac{P(\text{atributos}|\text{classe}) \cdot P(\text{classe})}{P(\text{atributos})}$$

Seja y a variável de classe (o que queremos prever) e $X = (x_1, x_2, \ldots, x_n)$ um vetor de n atributos. O objetivo é calcular a probabilidade de cada classe y_k para um dado conjunto de atributos X e selecionar a classe com a maior probabilidade.

A regra de decisão é, portanto, encontrar a classe y_k que maximiza a probabilidade a posteriori:

$$\hat{y} = \arg\max_k P(y_k|x_1, x_2, \ldots, x_n)$$

Usando o Teorema de Bayes, podemos reescrever a expressão como:

$$\hat{y} = \arg\max_k \frac{P(x_1, x_2, \ldots, x_n|y_k) \cdot P(y_k)}{P(x_1, x_2, \ldots, x_n)}$$

O termo $P(x_1, x_2, \ldots, x_n)$ no denominador é o mesmo para todas as classes. Como estamos interessados apenas em encontrar a classe que maximiza a probabilidade, podemos descartar o denominador, pois ele não afeta a ordem relativa das probabilidades. A fórmula simplificada se torna:

$$\hat{y} = \arg\max_k P(x_1, x_2, \ldots, x_n|y_k) \cdot P(y_k)$$

4 A Suposição "Ingênua"de Independência

Calcular a verossimilhança $P(x_1, x_2, \ldots, x_n|y_k)$ é computacionalmente complexo, pois exige um conjunto de dados massivo para estimar a probabilidade conjunta de todas as combinações de atributos.

É aqui que entra a suposição "ingênua"do Naive Bayes: **todos os atributos são condicionalmente independentes uns dos outros, dada a classe**.

Isso significa que, para uma classe y_k, o valor do atributo x_i não depende do valor de nenhum outro atributo x_j (onde $i \neq j$). Matematicamente, isso nos permite simplificar a verossimilhança da seguinte forma:

$$P(x_1, x_2, \ldots, x_n|y_k) = \prod_{i=1}^{n} P(x_i|y_k)$$

Esta suposição é o "pulo do gato"do algoritmo. Ela transforma um problema complexo de probabilidade conjunta em um cálculo simples de produtos de probabilidades individuais.

5 O Modelo Final do Naive Bayes

Substituindo a suposição de independência na nossa regra de decisão, obtemos a fórmula final do classificador Naive Bayes:

$$\hat{y} = \arg\max_k \left(P(y_k) \prod_{i=1}^{n} P(x_i|y_k) \right)$$

Para treinar o modelo, o algoritmo precisa apenas calcular as seguintes probabilidades a partir do conjunto de dados de treinamento:

1. **Probabilidade a priori da classe ($P(y_k)$):** É a frequência relativa de cada classe no conjunto de treinamento.

$$P(y_k) = \frac{\text{Número de amostras da classe } y_k}{\text{Número total de amostras}}$$

2. **Verossimilhança condicional de cada atributo ($P(x_i|y_k)$):** É a probabilidade de um atributo x_i ocorrer, dado que a amostra pertence à classe y_k. O cálculo depende do tipo de atributo (categórico ou numérico). Para atributos categóricos, é a frequência relativa. Para atributos numéricos, geralmente se assume uma distribuição (como a Gaussiana) e se calculam seus parâmetros (média e desvio padrão) para cada classe.

5.1 Lidando com o Problema do Zero Absoluto

Um problema prático ocorre se uma determinada combinação de atributo e classe nunca aparecer no conjunto de treinamento. Nesse caso, a probabilidade condicional $P(x_i|y_k)$ seria zero. Como o cálculo final é um produto, isso anularia toda a probabilidade a posteriori da classe.

Para evitar isso, utiliza-se uma técnica de suavização, como a **Suavização de Laplace (Laplace Smoothing)**. Adiciona-se um valor pequeno α (geralmente 1) ao numerador e $\alpha \cdot K$ ao denominador, onde K é o número de valores distintos que o atributo pode assumir.

A fórmula para a verossimilhança com suavização de Laplace se torna:

$$P(x_i|y_k) = \frac{\text{Contagem}(x_i, y_k) + \alpha}{\text{Contagem}(y_k) + \alpha \cdot K}$$

6 Conclusão

O Naive Bayes é um algoritmo de classificação robusto, rápido e fácil de implementar. Embora sua suposição de independência condicional raramente seja verdadeira em problemas do mundo real, o modelo frequentemente apresenta um desempenho excelente. Sua força reside na capacidade de fornecer uma base sólida para problemas de classificação, especialmente em domínios com alta dimensionalidade, como o processamento de linguagem natural. Compreender seus fundamentos matemáticos é essencial para qualquer cientista de dados ou entusiasta de Machine Learning.