Algorytm Naive Bayes

Klasyfikacja oparta na twierdzeniu Bayesa

Paweł Gliwny

Wydział Fizyki i Informatyki Stosowanej
Uniwersytet Łódzki

Eksploracja Danych

Problem klasyfikacji

Pytanie: Czy dziś zagramy w tenisa?

Dane:

- Cechy (X): Pogoda, Temperatura, Wiatr
- Klasa docelowa (Y): Gra w tenisa (TAK/NIE)

Naiwne podejście: Znajdź w historii dzień z identycznymi warunkami.

Problem: Co jeśli takiego dnia nie było w danych?

Eksplozja kombinatoryczna

Dlaczego szukanie dokładnych dopasowań nie działa?

Nasz przykład:

- Pogoda: 3 wartości
- Temperatura: 3 wartości
- Wiatr: 2 wartości
- Kombinacji: $3 \times 3 \times 2 = 18$

W praktyce:

- 5 cech po 3 wartości: $3^5 = 243$
- 10 cech po 3 wartości: $3^{10} = 59\,049$

Potrzebujemy uproszczenia! $\rightarrow$ Naive Bayes

Prawdopodobieństwo warunkowe

Definicja: Prawdopodobieństwo zdarzenia X, pod warunkiem że zaszło Y:

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)}$$

Intuicja: Zawężamy przestrzeń do przypadków, gdzie Y jest prawdą.

Przykład: $P(\text{Słońce} \mid \text{TAK}) = ?$
Wśród 8 dni gdy graliśmy, 3 były słoneczne:

$$P(\text{Słońce} \mid \text{TAK}) = \frac{3}{8} = 0.375$$

Uwaga: $P(\text{Słońce} \mid \text{TAK}) \neq P(\text{TAK} \mid \text{Słońce})$

Twierdzenie Bayesa – intuicja

Co mówi twierdzenie Bayesa?
Jak **aktualizować przekonania** w świetle nowych dowodów.

$$\underbrace{P(H|D)}_{\text{Co wiem PO}} = \frac{\overbrace{P(D|H)}^{\text{Jak dane pasują}} \cdot \overbrace{P(H)}^{\text{Co wiedziałem PRZED}}}{\underbrace{P(D)}_{\text{Jak częste są dane}}}$$

Przykład medyczny: Test na rzadką chorobę (1 na 10 000 osób), czułość 99%.
Masz wynik pozytywny. Czy jesteś chory?

- P(test+ | chory) = 0.99 – to wiemy z badań klinicznych
- P(chory | test+) = ? – to chcesz wiedzieć!

Odpowiedź: Około 1% – większość pozytywnych wyników to fałszywe alarmy, bo choroba jest bardzo rzadka (prior jest niski).

Twierdzenie Bayesa

Pozwala **odwrócić** prawdopodobieństwo warunkowe:

$$P(Y \mid X) = \frac{P(X \mid Y) \cdot P(Y)}{P(X)}$$

Symbol	**Nazwa**	**Znaczenie**	**Źródło**
$P(Y \mid X)$	posterior	to chcemy obliczyć	ze wzoru
$P(X \mid Y)$	likelihood	P(cecha $\mid$ klasa)	z danych
$P(Y)$	prior	częstość klasy	z danych
$P(X)$	evidence	częstość cechy	pomijamy!

$P(X)$ jest stałe dla wszystkich klas, więc wystarczy porównywać:

$$P(Y \mid X) \propto P(X \mid Y) \cdot P(Y)$$

Naive Bayes – kluczowa idea

Problem: Jak obliczyć $P(\text{Słońce}, \text{Zimno}, \text{Silny} \mid \text{TAK})$?

Rozwiązanie – założenie niezależności warunkowej:

$$P(X_1, X_2, \ldots, X_n \mid Y) \approx \prod_{i=1}^{n} P(X_i \mid Y)$$

W naszym przykładzie:

$$P(\text{Słońce}, \text{Zimno}, \text{Silny} \mid \text{TAK}) \approx P(\text{Słońce} \mid \text{TAK}) \cdot P(\text{Zimno} \mid \text{TAK}) \cdot P(\text{Silny} \mid \text{TAK})$$

Dlaczego "Naive" (naiwne)?
Zakładamy niezależność cech – to założenie jest prawie zawsze fałszywe, ale algorytm i tak działa dobrze!

Algorytm w 5 krokach

1. **Oblicz** $P(\text{cecha} \mid \text{klasa})$ dla każdej cechy osobno
2. **Pomnóż** prawdopodobieństwa:

$$\text{Score}(Y) = P(Y) \cdot \prod_i P(X_i \mid Y)$$

3. **Powtórz** dla wszystkich klas (TAK, NIE)
4. **Normalizuj:**

$$P(Y \mid X) = \frac{\text{Score}(Y)}{\sum_k \text{Score}(Y_k)}$$

5. **Wybierz** klasę z najwyższym prawdopodobieństwem

Dlaczego Naive Bayes działa mimo fałszywego założenia?

- Nie potrzebujemy **dokładnych** prawdopodobieństw – wystarczy poprawny **ranking** klas
- Błędy estymacji często się **znoszą**
- Mniej parametrów = mniejsze **overfitting**
- Szybki i **skalowalny** – złożoność $O(n \cdot d \cdot k)$
  gdzie: n = liczba próbek, d = liczba cech, k = liczba klas

Dane treningowe – gra w tenisa

Dzień	Pogoda	Temperatura	Wiatr	Gra?
1	Słońce	Ciepło	Słaby	TAK
2	Słońce	Ciepło	Silny	NIE
3	Pochmurno	Ciepło	Słaby	TAK
4	Deszcz	Umiarkowana	Słaby	TAK
5	Deszcz	Zimno	Słaby	TAK
6	Deszcz	Zimno	Silny	NIE
7	Pochmurno	Zimno	Silny	TAK
8	Słońce	Umiarkowana	Słaby	TAK
9	Słońce	Zimno	Słaby	TAK
10	Deszcz	Umiarkowana	Silny	TAK

Podsumowanie: 10 dni, $8 \times$ TAK, $2 \times$ NIE

Krok 1: Prawdopodobieństwa a priori

Obliczamy jak często występuje każda klasa:

$$P(\text{TAK}) = \frac{8}{10} = 0.80 \qquad P(\text{NIE}) = \frac{2}{10} = 0.20$$

Interpretacja:

- W 80% dni historycznie graliśmy w tenisa
- W 20% dni nie graliśmy

To są nasze **prior probabilities** – punkt startowy przed uwzględnieniem cech.

Krok 2: Prawdopodobieństwa warunkowe – Pogoda

Liczymy $P(\text{Pogoda} \mid \text{Klasa})$ dla każdej kombinacji:

Dla klasy TAK (8 dni):

- Słońce: 3 dni $\Rightarrow \frac{3}{8}$
- Pochmurno: 2 dni $\Rightarrow \frac{2}{8}$
- Deszcz: 3 dni $\Rightarrow \frac{3}{8}$

Dla klasy NIE (2 dni):

- Słońce: 1 dzień $\Rightarrow \frac{1}{2}$
- Pochmurno: 0 dni $\Rightarrow \frac{0}{2}$ (!)
- Deszcz: 1 dzień $\Rightarrow \frac{1}{2}$

$P(\text{Pogoda} \mid Y)$	TAK	NIE
Słońce	0.375	0.500
Pochmurno	0.250	0.000
Deszcz	0.375	0.500

Krok 2: Prawdopodobieństwa warunkowe – Temperatura i Wiatr

Temperatura:

$P(\text{Temp} \mid Y)$	TAK	NIE
Ciepło	$\frac{2}{8} = 0.25$	$\frac{1}{2} = 0.50$
Umiarkowana	$\frac{3}{8} = 0.375$	$\frac{0}{2} = 0.00$
Zimno	$\frac{3}{8} = 0.375$	$\frac{1}{2} = 0.50$

Wiatr:

$P(\text{Wiatr} \mid Y)$	TAK	NIE
Słaby	$\frac{6}{8} = 0.75$	$\frac{0}{2} = 0.00$
Silny	$\frac{2}{8} = 0.25$	$\frac{2}{2} = 1.00$

Uwaga: Widzimy kilka zer! To będzie problem...

Obserwacja: Gdy NIE graliśmy, zawsze był silny wiatr.

Krok 3: Predykcja dla nowego dnia

Nowy dzień: Słońce, Zimno, Silny wiatr – czy gramy?

Obliczamy Score dla każdej klasy:

$$\text{Score}(\text{TAK}) = P(\text{TAK}) \times P(\text{Słońce}|\text{TAK}) \times P(\text{Zimno}|\text{TAK}) \times P(\text{Silny}|\text{TAK})$$

$$= 0.80 \times 0.375 \times 0.375 \times 0.25 = 0.0281$$

$$\text{Score}(\text{NIE}) = P(\text{NIE}) \times P(\text{Słońce}|\text{NIE}) \times P(\text{Zimno}|\text{NIE}) \times P(\text{Silny}|\text{NIE})$$

$$= 0.20 \times 0.50 \times 0.50 \times 1.00 = 0.0500$$

Score(NIE) > Score(TAK) $\Rightarrow$ Predykcja: NIE gramy!

Krok 4: Normalizacja (opcjonalnie)

Chcemy prawdziwe prawdopodobieństwa (sumujące się do 1):

$$P(\text{TAK} \mid X) = \frac{\text{Score}(\text{TAK})}{\text{Score}(\text{TAK}) + \text{Score}(\text{NIE})} = \frac{0.0281}{0.0281 + 0.0500} = \frac{0.0281}{0.0781} \approx 0.36$$

$$P(\text{NIE} \mid X) = \frac{0.0500}{0.0781} \approx 0.64$$

Interpretacja:

- 36% szans, że zagramy
- 64% szans, że NIE zagramy

Predykcja: NIE (klasa z wyższym prawdopodobieństwem)

Problem zerowych prawdopodobieństw

Problem: Co jeśli kombinacja cechy i klasy nigdy nie wystąpiła?

Przykład: Brak dnia z Pogoda=Pochmurno i Gra=NIE

$$P(\text{Pochmurno} \mid \text{NIE}) = \frac{0}{2} = 0$$

Wtedy:

$$\text{Score}(\text{NIE}) = P(\text{NIE}) \cdot 0 \cdot \ldots = 0$$

Jedna "niewidziana" kombinacja zeruje całą klasę!
Brak danych $\neq$ niemożliwość zdarzenia

Laplace Smoothing (wygładzanie)

Idea: Dodajemy pseudo-liczbę α do każdej kategorii.

Bez smoothingu:

$$P(x \mid y) = \frac{\text{count}(x, y)}{\text{count}(y)}$$

Ze smoothingiem ($\alpha = 1$):

$$P(x \mid y) = \frac{\text{count}(x, y) + \alpha}{\text{count}(y) + \alpha \cdot k}$$

gdzie k = liczba kategorii dla danej cechy

Przykład (Pogoda ma 3 kategorie):

$$P(\text{Pochmurno} \mid \text{NIE}) = \frac{0+1}{2+3} = \frac{1}{5} = 0.20$$

W sklearn: `alpha=1.0` (domyślnie)

Kodowanie cech kategorycznych

Problem: Algorytmy ML nie rozumieją tekstu!

Metoda	**Zastosowanie**	**Przykład**
LabelEncoder	Zmienna docelowa (y)	TAK→1, NIE→0
OrdinalEncoder	Cechy porządkowe	Zimno→0, Umiark.→1, Ciepło→2
OneHotEncoder	Cechy nominalne	Słońce→[0,0,1]

Dla Naive Bayes: `CategoricalNB` traktuje wartości jako indeksy kategorii, nie jako wartości numeryczne – można użyć `LabelEncoder` dla wszystkich cech.

Przykład kodowania w sklearn

```
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder

# Zmienna docelowa
le = LabelEncoder()
y = le.fit_transform(['TAK', 'NIE', 'TAK'])  # [1, 0, 1]

# Cecha porzadkowa (z zachowaniem kolejnosci)
oe = OrdinalEncoder(categories=[['Zimno', 'Umiarkowana', 'Cieplo']])
temp = oe.fit_transform([['Zimno'], ['Cieplo']])  # [[0.], [2.]]

# Cecha nominalna
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(sparse_output=False)
pogoda = ohe.fit_transform([['Slonce'], ['Deszcz']])
# [[0, 1], [1, 0]]  # kolumny: Deszcz, Slonce
```

Warianty Naive Bayes w sklearn

Wariant	Typ danych	Przykład użycia
GaussianNB	Cechy ciągłe (float)	Temperatura w °C, waga
MultinomialNB	Zliczenia (int ≥ 0)	Liczba słów w tekście
CategoricalNB	Cechy kategoryczne	Pogoda, Wiatr
BernoulliNB	Cechy binarne (0/1)	Czy pada? Czy wieje?
ComplementNB	Niezbalansowane klasy	Gdy jedna klasa dominuje

Dla naszego przykładu z tenisem: `CategoricalNB`

Przykład: CategoricalNB

```
from sklearn.naive_bayes import CategoricalNB
from sklearn.preprocessing import LabelEncoder
import numpy as np

# Dane (zakodowane)
X = np.array([[2, 0, 1],   # Slonce, Zimno, Silny
[1, 1, 0],   # Pochmurno, Umiarkowana, Slaby
[0, 2, 1]])  # Deszcz, Cieplo, Silny
y = np.array([1, 1, 0])    # TAK, TAK, NIE

# Model
model = CategoricalNB(alpha=1.0)  # Laplace smoothing
model.fit(X, y)

# Predykcja
nowy_dzien = np.array([[2, 0, 1]])  # Slonce, Zimno, Silny
print(model.predict(nowy_dzien))        # [1] -> TAK
print(model.predict_proba(nowy_dzien))  # prawdopodobienstwa
```

Podsumowanie

Naive Bayes – kluczowe punkty:

- Oparty na twierdzeniu Bayesa: $P(Y \mid X) \propto P(X \mid Y) \cdot P(Y)$
- Zakłada niezależność cech (naiwne założenie)
- Działa dobrze mimo fałszywego założenia
- Szybki, skalowalny, odporny na overfitting

Pamiętaj:

- Laplace smoothing (`alpha`) – unikaj zerowych prawdopodobieństw
- Dobierz wariant do typu danych (Gaussian, Multinomial, Categorical, Bernoulli)
- Odpowiednio koduj cechy kategoryczne

Algorytm w pigułce

$$\hat{y} = \arg\max_{k} \left[P(Y_k) \cdot \prod_{i=1}^{n} P(X_i \mid Y_k) \right]$$

1. Oblicz P(cecha | klasa) dla każdej cechy
2. Pomnóż prawdopodobieństwa
3. Powtórz dla wszystkich klas
4. Normalizuj (opcjonalnie)
5. Wybierz klasę z max prawdopodobieństwem