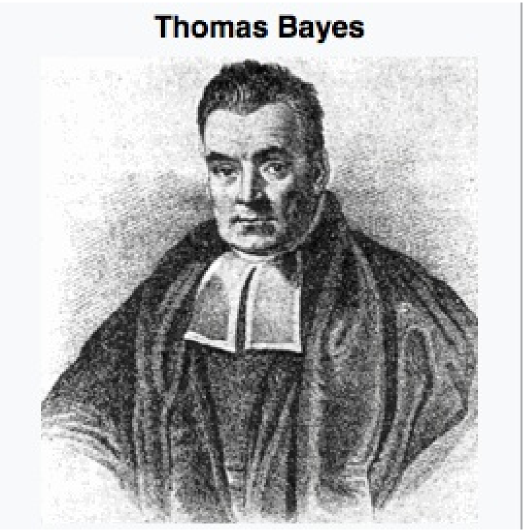
Naive Bayes



Lluís Talavera, 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

ETSEIB

A reminder of Bayes theorem

Given two events A and B,

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

- P(A) = probability of A ocurring (prior)
- P(B) = probability of B ocurring (evidence)
- P(B|A) = conditional probability of B given A (probability of B ocurring if a has ocurred, likelihood)
- P(A|B) = conditional probability of A given (probability of A ocurring if B has ocurred, posterior)

Naive Bayes

It is a conditional probability model. Given an example x represented by n features $(x_1,x_2,…,x_n)$ it assigns $P(y|x)$ for each possible class.

Using the Bayes theorem it can be decomposed as:

$$P(y|x) = \large\frac{P(y) \cdot P(x|y)}{P(x)}$$

We are only interested in the numerator because $P(x)$ is constant among all the classes. $P(y)$ is easy to compute. If we assume that every pair of features are **conditionally independent** given any class, estimation of the likelihood is easy:

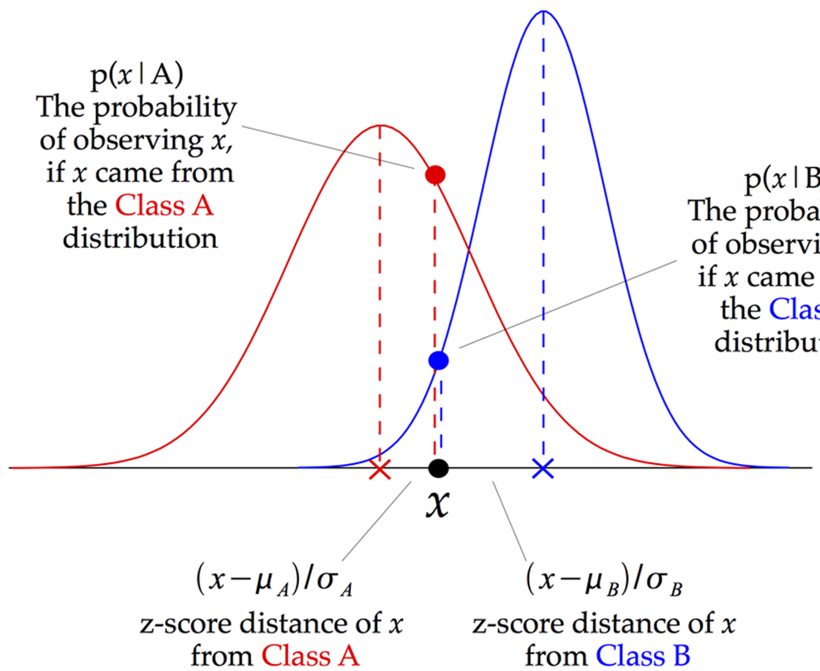
$$P(x|y) = \prod_{i=1}^n P(x_i|y) = P(x_1|y) \cdot P(x_2|y) \dots P(x_n|y)$$

The independence assumption may be too naive for real data!

Gaussian Naive Bayes

The likelihood of a value for column i is assumed to follow a Gaussian distribution:

$P(x_i|y) = \large \frac{1}{\sqrt{2 \pi \sigma_y}}\normalsize exp(-\frac{x_i - \mu_i}{2 \sigma^{2}_i})$



Naive Bayes algorithm

Training

Estimate $P(y=y_k)$ for each class label and $P(X|y=y_k)$ for each feature and class label from the training data.

(for Gaussian, estimate class conditional mean μ_{ik} and variance σ^{2}_{ik} for each feature i and class label k)

Prediction

Return the class of a new example $x = (x_1, x_2, \dots x_n)$ as

$\underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^n P(x_i|y)$

An example (I)

Brake?	Distance	Speed
Y	2.4	11.3
Y	3.2	70.2
N	75.7	72.7
N	2.8	15.2
?	**79.2**	**12.1**

Given a dataset with two features that represent the *Distance* to a corner of a player in a game and his *Speed*, determine if the player would brake or not.

Training the Gaussian NB model we obtain:

\$y\$	\$P(y)\$
Y	0.5
N	0.5

Prior probability

	Distance	Speed
\$\mu_Y\$	2.8	40.75
\$\mu_N\$	39.25	43.95
\$\sigma_Y^2\$	0.32	1734.605
\$\sigma_N^2\$	2657.205	1653.125

Parameters estimated to calculate the likelihood

An example (II)

Given the new example x:

Brake?	Distance	Speed
?	79.2	12.1

Use the probability density function for the Normal distribution with the estimated parameters to calculate the probability of the new example given each of the classes. For example:

$$P(speed=12.1|Y)=\frac{1}{\sqrt{2\cdot\pi\cdot 1734.605}} \cdot \exp\left(-\frac{(12.1-40.75)^2}{2\cdot 1734.605}\right)$$

$$=0.00756$$

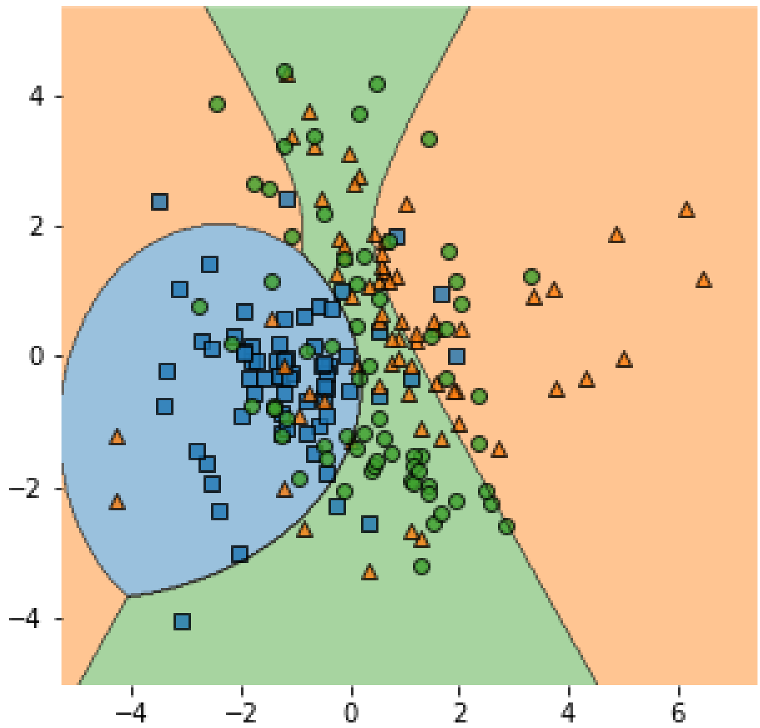
Prediction:

$$P(Y|x)=0.5\cdot 0.0\cdot 0.00756 = 0.0$$

$$P(N|x)=0.5\cdot 0.00573\cdot 0.00722 = \mathbf{0.00002}$$

NB decision boundaries

The decision boundaries can be linear (if variance is similar across the classes) or quadratic, as in the following example:



Some remarks about NB

- Fast training and prediction.
- The independence assumption often does not hold true in practice but, nevertheless, Naive Bayes may perform well!
- The naive assumption often leads to poor probabilities estimates, but it still can be a good classifier because it ranks the probabilites correctly.
- Not sensitive to irrelevant features.
- Correlated features can degrade performance.
- Imports with `sklearn`

```python
from sklearn.naive_bayes import GaussianNB
```