

# 1 Naive Bayes Classification

Let's say that we work with a dataset of  $n$  observations (rows) and  $m$  output classes where we want to classify  $n$  texts.

## 1.1 Definitions and Notations

Let  $T = \{x_1, x_2, \dots, x_n\}$  be the multiset of texts where every text  $x_i$  is defined by a multiset of words  $\{w_{i,1}, w_{i,2}, \dots, w_{i,k_i}\}$ . Note that the position of the texts in  $X$  does not matter.

We note

$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,m} \\ y_{2,1} & y_{2,2} & \dots & y_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & \dots & y_{n,m} \end{bmatrix} \quad (1)$$

the matrix of binary output values (explained variables)  $y_{i,j} \in \{0, 1\}$  for  $1 \leq i \leq n$  and  $1 \leq j \leq m$ .

Since the goal is to estimate  $Y$  because we are not supposed to know  $y_{i,j}$ , we note

$$\hat{Y} = \begin{bmatrix} \hat{y}_{1,1} & \hat{y}_{1,2} & \dots & \hat{y}_{1,m} \\ \hat{y}_{2,1} & \hat{y}_{2,2} & \dots & \hat{y}_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{n,1} & \hat{y}_{n,2} & \dots & \hat{y}_{n,m} \end{bmatrix} \quad (2)$$

the estimator matrix of  $Y$  where  $\hat{y}_{i,j} \in [0, 1]$  because we want to give a probability.

Between the estimated and the true values, there is generally a bias that we note

$$\epsilon = \begin{bmatrix} \epsilon_{1,1} & \epsilon_{1,2} & \dots & \epsilon_{1,m} \\ \epsilon_{2,1} & \epsilon_{2,2} & \dots & \epsilon_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{n,1} & \epsilon_{n,2} & \dots & \epsilon_{n,m} \end{bmatrix} \quad (3)$$

where  $\epsilon_{i,j} \in [-1, 1]$  because the bias may be negative or positive. If  $y_{i,j} = 1$  and the model estimated  $\hat{y}_{i,j} = 0.971$ , then the bias is positive because  $\epsilon_{i,j} = 1 - 0.97 = 0.03$ . However, if  $y_{i,j} = 0$  and  $\hat{y}_{i,j} = 0.12$ , then the bias is negative because  $\epsilon_{i,j} = 0 - 0.12 = -0.12$ .

We deduce the vectored equation

$$Y = \hat{Y} + \epsilon \quad (4)$$

where the operator  $+$  is the element-wise matrix addition.

Let  $f : \mathcal{T} \rightarrow \mathbb{M}_{n \times m}([0, 1])$  be a model defined by  $f(x) = \hat{Y}$  where the notation  $\mathbb{M}_{n \times m}([0, 1])$  means the set of matrix  $n$  by  $m$  for which each element is a real number in  $[0, 1]$ .

The goal is to find a model  $f$  such that the bias  $\epsilon$  is minimized when  $f$  is applied on  $x$ . Obtaining  $\epsilon = \mathbf{0}_{n \times m}$  means that the model  $f$  predict perfectly how the texts will be classified.

## 1.2 Theoretical Problem

Let  $C = \{c_1, c_2, \dots, c_m\}$  be the set of all output class labels. We define the following random variables:

- $c \in C$  representing an output class label;
- $x \in T$  representing a text.

For a given text  $x$ , in virtue of the Bayes theorem, we have

$$\mathbb{P}(c = c_j | x = x_i) = \frac{\mathbb{P}(x = x_i | c = c_j) \mathbb{P}(c = c_j)}{\mathbb{P}(x = x_i)}. \quad (5)$$

The goal of the Naive Bayes Classification is to find the output class label  $c$  that maximize the probability that a text  $x \in T$  maps to the output class label  $c_j$  knowing  $x_i$ . In other terms, this means that

$$c_{max} = \arg \max_{c \in C} \mathbb{P}(c = c_j | x = x_i) = \arg \max_{c \in C} \frac{\mathbb{P}(x = x_i | c = c_j) \mathbb{P}(c = c_j)}{\mathbb{P}(x = x_i)}. \quad (6)$$

We extract 2 assumptions that will simplify the equation (6):

1.  $\mathbb{P}(x = x_i)$  is the same for all output class labels and does not affect the argmax which is on  $c$ .
2. Two texts  $x_i, x_j \in T$  where  $i \neq j$  are independent. This implies that  $\mathbb{P}(x = x_i | c)$  is independent of  $\mathbb{P}(x = x_j | c)$ .

Applying the first property gives

$$c_{max} = \arg \max_{c \in C} \mathbb{P}(x | c) \mathbb{P}(c) \quad (7)$$

which can be written equivalently using the definition of  $T$  as

$$c_{max} = \arg \max_{c \in C} \mathbb{P}(x = x_1, x = x_2, \dots, x = x_n | c) \mathbb{P}(c). \quad (8)$$

Now, applying the second property in (8) gives

$$c_{max} = \arg \max_{c \in C} \mathbb{P}(c) \prod_{i=1}^n \mathbb{P}(x = x_i | c). \quad (9)$$

### 1.3 Bag of Words Model

Let  $W$  be the set of words contained in  $T$ . Take  $x_i = (w_{i,1}, w_{i,2}, \dots, w_{i,k_i}) \in \mathcal{T}$  a text containing  $k_i$  words where a word  $w_{i,j} \in W$ . We assume that a text cannot be empty meaning that  $\mathcal{T} \neq \emptyset$ .

We want to use the maximum likelihood estimator  $\hat{P}$  defined as the frequency of a word  $w_{i,j}$  among the  $n$  texts where  $1 \leq j \leq k_i$  knowing the output class label  $c_l \in C$ . The estimator  $\hat{P}$  estimates the likelihood function  $P(x_1, x_2, \dots, x_n; c) = \prod_{i=1}^n \mathbb{P}(x = x_i | c = c_l)$ .

To calculate that frequency, we have to calculate the ratio between the number of occurrences of the word  $w_{i,j}$  among the  $n$  texts, where the output class label is  $c_j$ , and the total number of words in the  $n$  texts where the output class label is  $c_j$ .

Let  $f : W \times C \rightarrow \mathbb{N}$  be a function defined as  $f(w_{i,j}, c_l) = z$  that returns the number of occurrences ( $z$ ) a word  $w_{i,j}$  is found among all texts classified as the output class label  $c_l$ .

Therefore, the maximum likelihood estimator of  $P(x_1, x_2, \dots, x_n; c)$  is defined as

$$\hat{P}(w_{i,j} \in x_i | c) = \frac{f(w_{i,j}, c) + 1}{\sum_{w \in W} f(w, c) + 1}. \quad (10)$$

We also need the maximum likelihood estimator of  $P(c = c_l) = \mathbb{P}(c = c_l)$  which is defined as the ratio between the number of texts classified as  $c_l$  and the number of texts  $n$ . Let  $T_c = \{x_i \in T : x_i \mapsto c_l\}$  be the set of all texts  $x_i$  classified as  $c_l$ .

We note  $|T_c|$  the cardinality of  $T_c$ . The estimator is defined as

$$\hat{P}(c = c_l) = \frac{|T_c|}{n}. \quad (11)$$

The reason behind the Laplace smoothing, adding 1 to the numerator and denominator of  $\hat{P}(w_{i,j} \in x_i | c)$ , is to handle the case when  $f(w_{i,j}, c) = 0$ . If a word  $w_{i,j}$  is not found for a given output class label  $c_l$ , then  $\hat{P}(w_{i,j} \in x_i | c) = 0$ . Having only one case like this without adding 1 causes

$$\hat{P}(c) \prod_{i=1}^n \hat{P}(w_{i,j} \in x_i | c) = 0. \quad (12)$$

### 1.4 Example

In this example, we want to classify texts as toxic or non toxic. Suppose that we have the train dataset 1 where the texts have already been cleaned.

We have to predict if the text *shit language yourself hell fuck shit* is toxic or not.

Table 1: Train Dataset

| Text  | Is Toxic |
|---|----------|
| fuck fuck fuck shit shit                          | 1        |
| explanation natural processing<br>language matter | 0        |
| hell fuck die mother fuck shit                    | 1        |
| block pollution environment<br>climate natural    | 0        |
| mother fuck stupid piece shit                     | 1        |

From the dataset 1 including the text to classify, we set  $T = \{x_1, x_2, x_3, x_4, x_5, x_t\}$  as

$$\begin{aligned}
x_1 &= \{fuck, fuck, fuck, shit, shit\} \\
x_2 &= \{explanation, natural, processing, language, matter\} \\
x_3 &= \{hell, fuck, die, mother, fuck, shit\} \\
x_4 &= \{block, pollution, environment, climate, natural, mother\} \\
x_5 &= \{mother, fuck, stupid, piece, shit\} \\
x_t &= \{shit, language, yourself, hell, fuck, shit\}.
\end{aligned}$$

where  $x_t$  is the text to classify. The output classes are  $Y = (1, 0, 1, 0, 1)$ .

Let  $W = \{fuck, shit, explanation, natural, processing, language, matter, hell, die, mother, block, pollution, environment, climate, stupid, piece, yourself\}$  be the set of words used in  $T$ . We have  $|W| = 17$ . Let  $W_t$  be the multiset of words in texts classified as toxic and  $W_n$  the multiset of words in texts classified as non toxic. Thus, we have  $|W_t| = 16$  and  $|W_n| = 11$ .

Let  $c \in C = \{\text{"toxic"}, \text{"non toxic"}\}$  be the random variable representing

an output class label and  $w \in W$  the random variable representing a word.

$$\begin{aligned}
\hat{P}(w = \text{"fuck"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{6 + 1}{16 + 17} = \frac{7}{33} = 0.2121 \\
\hat{P}(w = \text{"shit"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{4 + 1}{16 + 17} = \frac{5}{33} = 0.1515 \\
\hat{P}(w = \text{"hell"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{1 + 1}{16 + 17} = \frac{2}{33} = 0.0606 \\
\hat{P}(w = \text{"die"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{1 + 1}{16 + 17} = \frac{2}{33} = 0.0606 \\
\hat{P}(w = \text{"mother"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{2 + 1}{16 + 17} = \frac{3}{33} = 0.0909 \\
\hat{P}(w = \text{"stupid"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{1 + 1}{16 + 17} = \frac{2}{33} = 0.0606 \\
\hat{P}(w = \text{"piece"}|c = \text{"toxic"}) &= \frac{f(w, c) + 1}{|W_t| + |W|} = \frac{1 + 1}{16 + 17} = \frac{2}{33} = 0.0606 \\
\hat{P}(w = \text{"mother"}|c = \text{"non toxic"}) &= \frac{f(w, c) + 1}{|W_n| + |W|} = \frac{1 + 1}{11 + 17} = \frac{2}{28} = 0.0714 \\
\hat{P}(w = \text{"natural"}|c = \text{"non toxic"}) &= \frac{f(w, c) + 1}{|W_n| + |W|} = \frac{2 + 1}{11 + 17} = \frac{3}{28} = 0.1071
\end{aligned}$$

Note that for a word  $w \in \{\text{explanation, processing, language, matter, block, pollution, environment, climate}\}$ , the probability is  $\hat{P}(w|c = \text{"non toxic"}) = 0.0714$  and  $\hat{P}(w|c = \text{"toxic"}) = 0.0303$ .

Since the dataset contains 2 texts classified as toxic and 3 texts as non toxic, we have

$$\begin{aligned}
\hat{P}(c = \text{"toxic"}) &= \frac{3}{5} = 0.6 \\
\hat{P}(c = \text{"non toxic"}) &= \frac{2}{5} = 0.4.
\end{aligned}$$

Let's predict in which output class label  $x_t$  is classified. We use the equation (10).

$$\begin{aligned}
\hat{P}(c = \text{"toxic"}|x = x_t) &= 0.6 \times 0.1515^2 \times 0.0303 \times 0.0303 \times 0.0606 \times 0.2121 \\
&= 0.000000163 \\
\hat{P}(c = \text{"non toxic"}|x = x_t) &= 0.4 \times 0.0357^2 \times 0.0714 \times 0.0357 \times 0.0357 \times 0.0357 \\
&= 0.000000002
\end{aligned}$$

It follows that  $x_t$  is classified as a toxic text.