Naive Bayes Classifier

Julius Kasiske

October 2022

1. Introduction

Bayesian statistics have vast applications across multiple domains, one of which is Machine Learning, where Bayesian Classifiers are for instance used in spam detection software. Knowing when to apply Bayesian Learners requires an understanding of their functionalities and mathematical derivation. Hence, the following sections will explore the most easily understood probabilistic classifier, Naive Bayes. For that, Bayes Theorem will be rehearsed as the basis for the learner and its transfer to Machine Learning contexts will be substantiated.

2. Bayes Theorem

2.1 Intuition

When searching for `Bayes Theorem` online, the initial formula is found rather quickly. For those who did not search yet, it goes as follows:

$$P(A|B) = \frac{P(B|A)}{P(B)} \cdot P(A) \tag{1}$$

Bayes Theorem motivates lots of further applications, one of which is Bayesian Classifiers like Naive Bayes. So it is worth it to understand some base level intuition regarding why that equation actually holds. One approach to provide said intuition starts with understanding the basic rational behind conditional probabilities, which looks as follows:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2}$$

The intuition here is rather simple. When we want to find the probability of A given that B is true, then for that to happen, B must be true in the first place, i.e. $P(B)$ is in the denominator. And now, when we know B is true, we want to subset $P(B)$ by the cases in which A is also true, i.e. $P(A \cap B)$ is in the numerator.

As is rather obvious, the same relationship holds in reversed order, i.e.:

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \tag{3}$$

Note that the set intersection ($\cap$) is a commutative operation, i.e. $A \cap B = B \cap A$. With this in mind, motivating Bayes Theorem is now solely a matter of substituting equation (2) and (3).

We first solve (3) for $P(B \cap A)$

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

Plugging that result into (2) then yields Bayes Theorem:

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

2.2 Transfer to Machine Learning

As one might have guessed by now, by its definition, Bayes Theorem is applicable when we know what happened (B) and we want the probability that another event occurred, given that prior event, i.e. $P(A|B)$. Translated into a Machine Learning contexts, we could ask what the probability is that our response variable y has a certain value, given our set of explanatory variables X. Bayes Theorem could thus be given as:

$$P(y|X) = \frac{P(X|y)}{P(X)} \cdot P(y) \tag{4}$$

2.3 Bayesian Nomenclature

Researching Bayesian Classifiers often returns the use of words like `posterior`, `prior` or `evidence`. To prevent confusion, the following section aims at clarifying said nomenclature alongside other terms.

For this, it helps to view Bayesian problems to be defined within a period of time (diacronic view), where we start off with having some evidence, usually our data X, and formulate a hypothesis, usually our outcome variable y, based on our evidence. As an example, imagine we collected data on height, weight and hair length and hypothesize if the person is male or female. Then our data (explanatory variables) is known beforehand, it thus is our `evidence`, while our response variable (y) being either male or female is our `hypothesis`. We formulate our hypothesis $P(y|X)$ after we have gathered our evidence X.

In our diacronic view, we argue that before we compute the actual probability of the hypothesis, we make a prior estimation, which we call `prior`. That prior is usually $P(y)$, because we can use it to generalize statements about $P(y|X)$. In our gender classification example from above, if for instance, we want the probability that the person is male given his specific height, weight and hair length, it might be a good first guess to ask for the probability that the person is male in general. We thus ask for the proportion of males in our data.

If our first guess $P(y)$ is called the prior, then it only makes sense to call the final result of Bayes Theorem the `posterior` ($P(y|X)$), which leaves one last definition - that of `likelihood`.

In statistics, the likelihood - unlike the probability - is the chance to observe the data, given that we got our result. In our example, the likelihood would be $P(X|y)$. Bayes Theorem can thus be summarized as follows:

$$posterior = \frac{likelihood}{evidence} \cdot prior \tag{5}$$

3. Naive Bayes Classifier

3.1 Notation

To understand what is under the hood of Naive Bayes, we first need to refine the mathematical notation.

Let X be our set of input vectors (columns in the data matrix), i.e. a set of n-tuples x_i, where $i \in [1, r]$. Every vector x_i is of length n, s.t. $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k}, \ldots, x_{i,n})$. Each row in our data matrix is denoted by $h_k = (h_{k,1}, \ldots, h_{k,i}, \ldots, h_{k,r})$, where $k \in [1, n]$.

Let our response variable $y = (y_1, \ldots, y_k, \ldots, y_n)$ be categorical, where each $y_k \in C, \forall k \in [1, n]$ and $C = \{c_1, \ldots, c_l, \ldots, c_m\}$.

3.2 Objective Function and Assumption of Independence

For every row in our data, the idea now becomes to choose a category c_l for the response variable y, such that the probability of y_k being c_l is maximize, which we can formulate using Bayes Theorem as follows:

$$\underset{c \in C}{\operatorname{argmax}}\, P(y_k = c|h_k) = \underset{c \in C}{\operatorname{argmax}}\, \frac{P(h_k|y_k = c)}{P(h_k)} \cdot P(y_k = c) \tag{6}$$

Note that $P(h_k)$ is independent of the choice of c, it will thus not affect our maximizer, we can hence leave it out of the objective function for simplicity. Also note that $P(y_k = c)$ can be easily approximated, using relative frequencies of the response variable y. That leaves the numerator of the above fraction, namely $P(h_k|y_k = c)$. Note that h_k is actually a row vector of length r. Computing the probability for r different values, given what y_k is, turns out to be daunting. Theoretically, one could try to use relative frequencies to approximate that probability, like we can do with $P(y_k = c)$. However, if every expalanatory variables x is assumed to be categorical with j distinct possible categories, then the number of different rows h that exist is j^r. Assuming $j = 5$ and $r = 10$, i.e. ten explanatory variables each have five categories, then there exist almost ten million different possibilities for how any given row vector h_k can look like. In order to build relative frequencies for all of those, one would have to have at least ten million rows in the dataset, otherwise, not even every single possibility shows up in the data. So the approach of relative frequencies is impractical, if even feasible.

To circumvent that problem, one simplifying assumption needs to be made. If we assume that all explanatory variables in X are independent of one another, then we can look at probabilities for each attribute separately, instead of having to consider the probability of all attributes as a combination of values altogether. If random variables are independent, their joint probability is solely the multiplication of each singular probability.

Making the independence assumption thus transforms equation (6) into the following maximization problem:

$$\underset{c \in C}{\operatorname{argmax}} \prod_{i=1}^{r} [P(h_{k,i}|y = c)] \cdot P(y_k = c) \tag{7}$$

3.3 Getting the Probabilities

We now devised the final maximization problem and have briefly talked about how to compute both elements of the multiplication. To rewind, the two elements are Likelihood and Prior:

$$\underset{c \in C}{\operatorname{argmax}} \underbrace{\prod_{i=1}^{r} [P(h_{k,i}|y = c)]}_{\text{Likelihood}} \cdot \underbrace{P(y_k = c)}_{\text{Prior}} \tag{8}$$

In order for predictions to be made, both terms have to be turned into actual numbers. The following sections will detail how that is done.

3.3.1 Likelihood

Getting the Likelihood requires different methods depending on if the variables are discrete or continuous.

3.3.1.1 Multinomial Naive Bayes In the discrete case, the classifier is also referred to as Multinomial Naive Bayes, because features follow a multinomial distribution. The multinomial distribution, its probability mass function and derivation is however not crutial for the context of Naive Bayes. It is important to know that the likelihood is approximated by relative frequencies in the multinomial case. To illustrate this point, consider the following oversimplified example of spam detection - a very common application for Bayesian Classifiers. The following data matrix stores email texts in each row and gives an output for spam or not:

```
##    Word_1 Word_2  Word_3 Spam
## 1:     Hi Friend     Bye    N
## 2:  Hello  Buddy Regards    N
## 3:  Money    Act    Fast    Y
## 4:   Fast  Money     Act    Y
## 5:     Hi  Buddy     Bye    N
## 6:  Hello Friend     Bye    N
## 7:     Hi  Buddy Regards    N
```

To be in line with the above notation, note that in our example, $C = \{Y, N\}$ and $r = 3$. In order to classify future emails, we need to compute the relative frequencies for each of the three features. For instance, for $c = N$ and for the first word, $P(\text{Word\_1} = \text{“Hi”} \mid N) = \frac{3}{5}$, because there are five non-spam observations, out of which three have “Hi” as their first word. In its training phase, the algorithm does the same for all words in each column, then does that for each column and all that for each element in C.

3.3.1.2 Gaussian Naive Bayes The discrete case of multinomial distributions is rather simple to understand. Things turn slightly more complicated when explanatory variables are not discrete but continuous. Continuous random variables are characterized by their ability to attain an infinite amount of different values. Variables like body height, weight and hair length are usually considered continuous, because each can have infinitely granular gradations. That implies that relative frequencies of continuous random variables will always be zero, because the sample space (usually denoted by Ω) has infinitely many elements.

We usually solve this problem by assuming pre-parameterized distributions, like the normal distribution and use its density function to get the likelihoods. To recap, for any value x of the continuous random variable X, the density function of the normal distribution is defined as follows:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{9}$$

3.3.2 Prior

Attentive readers might have observed that the above problem formulation has one inherent weakness. Namely, since we calculate the product above for every possible value that y may attain, for every row, there exists the possibility of one or many conditional relative frequencies being zero, which propagates through the entire multiplication and turns the final product to zero. For instance, if we observe a male that weighs 50kg, is 169cm tall and has a hair length of 1cm, then, in order to make predictions about the gender, we also have to evaluate the probability that this person is a female for comparability. However, there probably is no female in our data set with hair length of 1cm. Thus, the conditional frequency of observing that hair length, given the person is female, is zero, turning the overall probability to zero, even when considering the other attributes that would rather indicate a female person. At its root, the problem is that we only estimate probabilities using frequencies, and these estimations might be wrong. To solve this problem, we add one count for all substrata, making it impossible to obtain relative frequencies of exactly zero. Doing so additionally stabilizes the overall model's estimates.

```
##    Hair_Length Height Weight Gender
## 1:          20    166     56      F
## 2:           5    189     79      M
## 3:           3    176     72      M
## 4:          32    165     60      F
```