$$P(H \backslash E) = \frac{P(E \backslash H) * P(H)}{P(E)}$$

Likelihood of the Evidence given that the Hypothesis is True

Prior Probability of the Hypothesis

Prior probability of the Hypothesis given that the Evidence is True

Prior probability that the evidence is True

Bayes' Theorem: The formula

$$\underbrace{P(A|B)}_{\text{Posterior}} = \frac{\underbrace{P(B|A)}_{\text{Likelihood}} * \underbrace{P(A)}_{\text{Prior}}}{\underbrace{P(B)}_{\text{Evidence}}}$$

It's the probability of event B given A has occurred

The prior probability of event A

It's the probability of event A given B was observed

The probability of observing B

@akshay_pachaar

Prior probability: The general belief!

☔ P(Rain) = 40%

It's the probability of an event before new evidence is taken into account.

It represents what is known about the event's likelihood before observing any new data or information.

@akshay_pachaar

The evidence!

☁️ P(Clouds) = 50%

Let's assume it's 50% or 0.50 for simplicity.

This includes all scenarios - both when it rains and when it does not.

@akshay_pachaar

The likelihood!

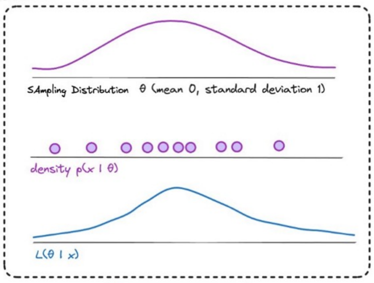
🌧️ P(Clouds|Rain) = 80% = L(Rain|Clouds)

The probability of observing clouds given that it's raining!

It can also be interpreted as the Likelihood it rains given clouds are there L(Rain|Clouds)!

Although it sounds similar to P(Rain|Clouds), the likelihood function here is used in Bayes' Theorem to weigh the evidence (clouds) in support of our hypothesis (rain), rather than calculating a straightforward probability.

Likelihood function



@akshay_pachaar

Posterior probability: Updating our belief!

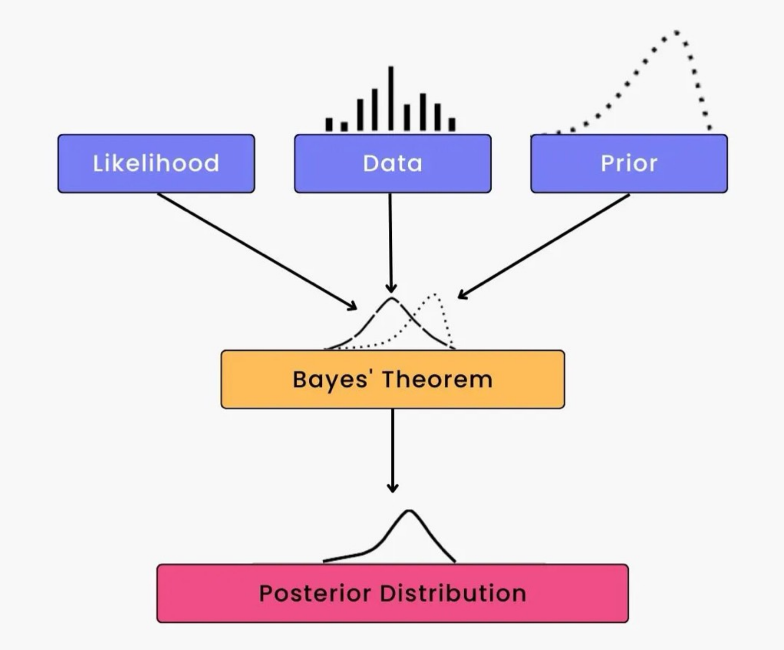
P(☔|☁️)

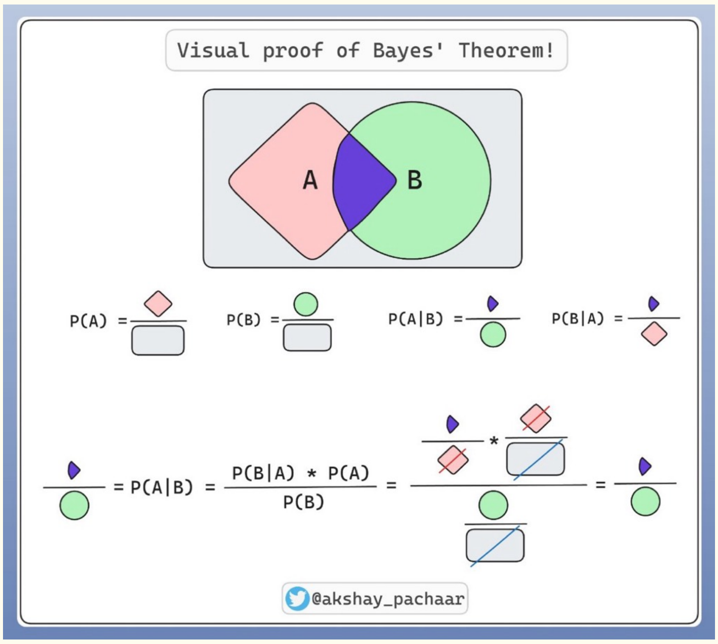
$$P(Rain|Cloud) = \frac{P(Cloud|Rain) * P(Rain)}{P(Cloud)}$$

$$= \frac{0.8 * 0.4}{0.5} = 0.64$$

Observe that our updated probability rises from 0.40 to 0.64, based on the new evidence!

@akshay_pachaar





$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features x_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C_k, x_1, \ldots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \ldots, x_n) &= p(x_1, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\ p(x_2, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\ p(x_2 \mid x_3, \ldots, x_n, C_k)\ p(x_3, \ldots, x_n, C_k) \\ &= \cdots \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\ p(x_2 \mid x_3, \ldots, x_n, C_k) \cdots p(x_{n-1} \mid x_n, C_k)\ p(x_n \mid C_k)\ p(C_k) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that all features in $\mathbf{x}$ are mutually independent, conditional on the category C_k. Under this assumption,

$$p(x_i \mid x_{i+1}, \ldots, x_n, C_k) = p(x_i \mid C_k)\ .$$

Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k \mid x_1, \ldots, x_n) \propto\ & p(C_k, x_1, \ldots, x_n) \\ &= p(C_k)\ p(x_1 \mid C_k)\ p(x_2 \mid C_k)\ p(x_3 \mid C_k)\ \cdots \\ &= p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,, \end{aligned}$$

where $\propto$ denotes proportionality since the denominator $p(\mathbf{x})$ is omitted.

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

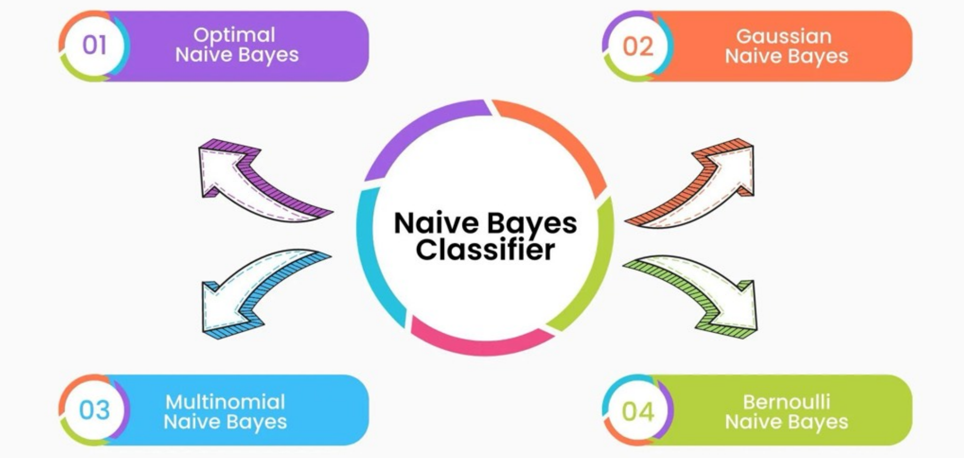
$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z}\, p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

where the evidence $Z = p(\mathbf{x}) = \sum_k p(C_k)\ p(\mathbf{x} \mid C_k)$ is a scaling factor dependent only on $x_1, \ldots, x_n$, that is, a constant if the values of the feature variables are known.

Constructing a classifier from the probability model [edit]

The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable so as to minimize the probability of misclassification; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k \in \{1, \ldots, K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$



Gaussian Naive Bayes

It is a straightforward algorithm used when the attributes are continuous. The attributes present in the data should follow the rule of Gaussian distribution or normal distribution. It remarkably quickens the search, and under lenient conditions, the error will be two times greater than Optimal Naive Bayes.

Optimal Naive Bayes

Optimal Naive Bayes selects the class that has the greatest posterior probability of happenings. As per the name, it is optimal. But it will go through all the possibilities, which is very slow and time-consuming.

Bernoulli Naive Bayes

Bernoulli Naive Bayes is an algorithm that is useful for data that has binary or boolean attributes. The attributes will have a value of yes or no, useful or not, granted or rejected, etc.

Multinominal Naive Bayes

Multinominal Naive Bayes is used on documentation classification issues. The features needed for this type are the frequency of the words converted from the document.

Bernoulli naive Bayes [edit]

In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks,[9] where binary term occurrence features are used rather than term frequencies. If x_i is a Boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class C_k is given by:[9]

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

where p_{ki} is the probability of class C_k generating the term x_i. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

Gaussian naive Bayes [edit]

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. For example, suppose the training data contains a continuous attribute, x. The data is first segmented by the class, and then the mean and variance of x is computed in each class. Let μ_k be the mean of the values in x associated with class C_k, and let σ_k^2 be the Bessel corrected variance of the values in x associated with class C_k. Suppose one has collected some observation value v. Then, the probability *density* of v given a class C_k, i.e., $p(x = v \mid C_k)$, can be computed by plugging v into the equation for a normal distribution parameterized by μ_k and σ_k^2. Formally,

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Another common technique for handling continuous values is to use binning to discretize the feature values and obtain a new set of Bernoulli-distributed features. Some literature suggests that this is required in order to use naive Bayes, but it is not true, as the discretization may throw away discriminative information.[3]

Sometimes the distribution of class-conditional marginal densities is far from normal. In these cases, kernel density estimation can be used for a more realistic estimate of the marginal densities of each class. This method, which was introduced by John and Langley,[8] can boost the accuracy of the classifier considerably.[11][12]

Multinomial naive Bayes [edit]

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial $(p_1, \dots, p_n)$ where p_i is the probability that event *i* occurs (or K such multinomials in the multiclass case). A feature vector $\mathbf{x} = (x_1, \dots, x_n)$ is then a histogram, with x_i counting the number of times event *i* was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram $\mathbf{x}$ is given by:

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_{i=1}^n x_i)!}{\prod_{i=1}^n x_i!} \prod_{i=1}^n {p_{ki}}^{x_i} \quad \text{where } p_{ki} := p(x_i \mid C_k).$$

The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space:[13]

$$\begin{aligned}
\log p(C_k \mid \mathbf{x}) &\varpropto \log\left(p(C_k) \prod_{i=1}^n {p_{ki}}^{x_i}\right) \\
&= \log p(C_k) + \sum_{i=1}^n x_i \cdot \log p_{ki} \\
&= b + \mathbf{w}_k^\top \mathbf{x}
\end{aligned}$$

where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$. Estimating the parameters in log space is advantageous since multiplying a large number of small values can lead to significant rounding error. Applying a log transform reduces the effect of this rounding error.

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero, because the probability estimate is directly proportional to the number of occurrences of a feature's value. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

Rennie *et al.* discuss problems with the multinomial assumption in the context of document classification and possible ways to alleviate those problems, including the use of tf–idf weights instead of raw term frequencies and document length normalization, to produce a naive Bayes classifier that is competitive with support vector machines.[13]

Training [edit]

Example training set below.

Person	height (feet)	weight (lbs)	foot size (inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

The classifier created from the training set using a Gaussian distribution assumption would be (given variances are *unbiased* sample variances):

Person	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033×10^{-2}	176.25	1.2292×10^{2}	11.25	9.1667×10^{-1}
female	5.4175	9.7225×10^{-2}	132.5	5.5833×10^{2}	7.5	1.6667

The following example assumes equiprobable classes so that P(male)= P(female) = 0.5. This prior probability distribution might be based on prior knowledge of frequencies in the larger population or in the training set.

Testing [edit]

Below is a sample to be classified as male or female.

Person	height (feet)	weight (lbs)	foot size (inches)
sample	6	130	8

In order to classify the sample, one has to determine which posterior is greater, male or female. For the classification as male the posterior is given by

$$\text{posterior (male)} = \frac{P(\text{male})\, p(\text{height} \mid \text{male})\, p(\text{weight} \mid \text{male})\, p(\text{foot size} \mid \text{male})}{evidence}$$

For the classification as female the posterior is given by

$$\text{posterior (female)} = \frac{P(\text{female})\, p(\text{height} \mid \text{female})\, p(\text{weight} \mid \text{female})\, p(\text{foot size} \mid \text{female})}{evidence}$$

The evidence (also termed normalizing constant) may be calculated:

$$\begin{aligned}\text{evidence} = {} & P(\text{male})\, p(\text{height} \mid \text{male})\, p(\text{weight} \mid \text{male})\, p(\text{foot size} \mid \text{male}) \\ & + P(\text{female})\, p(\text{height} \mid \text{female})\, p(\text{weight} \mid \text{female})\, p(\text{foot size} \mid \text{female})\end{aligned}$$

However, given the sample, the evidence is a constant and thus scales both posteriors equally. It therefore does not affect classification and can be ignored. The probability distribution for the sex of the sample can now be determined:

$$P(\text{male}) = 0.5$$

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789,$$

where $\mu = 5.855$ and $\sigma^2 = 3.5033 \cdot 10^{-2}$ are the parameters of normal distribution which have been previously determined from the training set. Note that a value greater than 1 is OK here – it is a probability density rather than a probability, because *height* is a continuous variable.

$$p(\text{weight} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(130-\mu)^2}{2\sigma^2}\right) = 5.9881 \cdot 10^{-6}$$

$$p(\text{foot size} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(8-\mu)^2}{2\sigma^2}\right) = 1.3112 \cdot 10^{-3}$$

$$\text{posterior numerator (male)} = \text{their product} = 6.1984 \cdot 10^{-9}$$

$$P(\text{female}) = 0.5$$

$$p(\text{height} \mid \text{female}) = 2.23 \cdot 10^{-1}$$

$$p(\text{weight} \mid \text{female}) = 1.6789 \cdot 10^{-2}$$

$$p(\text{foot size} \mid \text{female}) = 2.8669 \cdot 10^{-1}$$

$$\text{posterior numerator (female)} = \text{their product} = 5.3778 \cdot 10^{-4}$$

Since posterior numerator is greater in the female case, the prediction is that the sample is female.

Advantages of a Naive Bayes Classifier

Here are some advantages of the Naive Bayes Classifier:

- It doesn't require larger amounts of training data.
- It is straightforward to implement.
- Convergence is quicker than other models, which are discriminative.
- It is highly scalable with several data points and predictors.
- It can handle both continuous and categorical data.
- It is not sensitive to irrelevant data and doesn't follow the assumptions it holds.
- It is used in real-time predictions.

Disadvantages of a Naive Bayes Classifier

The disadvantage of the Naive Bayes Classifier are as below:

- The Naive Bayes Algorithm has trouble with the 'zero-frequency problem'. It happens when you assign zero probability for categorical variables in the training dataset that is not available. When you use a smooth method for overcoming this problem, you can make it work the best.
- It will assume that all the attributes are independent, which rarely happens in real life. It will limit the application of this algorithm in real-world situations.
- It will estimate things wrong sometimes, so you shouldn't take its probability outputs seriously.