In the supervised learning algorithms known as naïve Bayes methods, the Bayes’ theorem is applied while every pair of features is naively assumed to be independent. As such, the theorem states: (write the eqn) Where Y is a class variable and X1 through Xn is the dependent feature vectors.

Since the naïve assumption is mathematically represented as (write eqn), the relationship could be boiled down to (write eqn) for all i.

Given the input, P(x1,…,xn) is a constant. Subsequently, this classification rule could be applied (write eqn).

Following this logic, Maximum A posteriori (MAP) could be employed to estimate the relative frequency of the class Y in a given training set, P(y) , as well as P(xi|y).

How the distribution of P(xi|y) is assumed to be is the main distinction between the different classifiers based on the naïve Bayes methods. To give an estimate of the required parameters, these classifiers require only a small training data set. As such, they were successfully employed in several situations despite their apparently over-simplified assumptions. In fact, this simplicity of the naïve assumption reduces the dimensionality problem by decoupling class conditional feature distributions. This makes naïve Bayes learners and classifiers much faster than other more sophisticated methods. One drawback, however, is being a poor estimator; the outcomes from predict\_proba are not very accurate.

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of independence between every pair of features. Given a class variable http://scikit-learn.org/stable/_images/math/276f7e256cbddeb81eee42e1efc348f3cb4ab5f8.png and a dependent feature vector _1 through _n, Bayes’ theorem states the following relationship:

(y \mid x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots x_n \mid y)}
                

Using the naive independence assumption that

(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),

for all http://scikit-learn.org/stable/_images/math/df0deb143e5ac127f00bd248ee8001ecae572adc.png, this relationship is simplified to

(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}
           

Since (x_1, \dots, x_n) is constant given the input, we can use the following classification rule:

(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)

\Downarrow


and we can use Maximum A Posteriori (MAP) estimation to estimate (y) and (x_i \mid y); the former is then the relative frequency of class http://scikit-learn.org/stable/_images/math/276f7e256cbddeb81eee42e1efc348f3cb4ab5f8.png in the training set.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of (x_i \mid y).

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. (For theoretical reasons why naive Bayes works well, and on which types of data it does, see the references below.)

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

On the flip side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.