# Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption that the presence of a particular feature in a class is unrelated to the presence of any other feature. This algorithm is mostly used in text classification and problems having multiple classes.

## 1. Bayes Theorem

Given a hypothesis H and evidence E, Bayes theorem states that the relationship between the probability of the hypothesis H before getting the evidence E (P(H)) and the probability of the hypothesis H after getting the evidence E (P(H/E) is:

$$\boldsymbol{P}(H|E) = \frac{\boldsymbol{P}(E|H)}{\boldsymbol{P}(E)}\boldsymbol{P}(H)$$

Here P(H/E) is posterior probability, P(H) is prior probability, P(E/H)/P(E) is likelihood ratio.

After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the maximum a posteriori (MAP) hypothesis.

MAP(h) = max(P(H|E)) = max(P(E|H) * P(H)/P(E)) ~ max(P(E|H) * P(H))

Here P(E) is a normalizing term which allows us to calculate the probability. We drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize.

## 2. Naive Bayes Algorithm for Machine Learning

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. Given a class variable y and a dependent feature vector X($x_1$ through $x_n$), Bayes' theorem states the following relationship:

$$\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x}_1,...\boldsymbol{x}_n) = \frac{\boldsymbol{P}(\boldsymbol{y})\boldsymbol{P}(\boldsymbol{x}_1,...\boldsymbol{x}_n|\boldsymbol{y})}{\boldsymbol{P}(\boldsymbol{x}_1,...\boldsymbol{x}_n)}$$

Using the naïve independence assumption that

$$\boldsymbol{P}(\boldsymbol{x}_i|\boldsymbol{y},\boldsymbol{x}_1,...\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},...\boldsymbol{x}_n) = \boldsymbol{P}(\boldsymbol{x}_i|\boldsymbol{y})$$

For all i, this relationship is simplified to:

$$\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x}_1,...\boldsymbol{x}_n) = \frac{\boldsymbol{P}(\boldsymbol{y})\prod_{i=1}^{n}\boldsymbol{P}(\boldsymbol{x}_i|\boldsymbol{y})}{\boldsymbol{P}(\boldsymbol{x}_1,...\boldsymbol{x}_n)}$$

Since $P(x_1,...x_n)$ is constant given the input, the following classification rule can be used for making prediction:

$$P(y|x_1,...x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$predicted\_y = \max P(y)\prod_{i=1}^{n} P(x_i|y)$$

Then we use MAP estimation to estimate $P(y|x_1,...x_n)$. **The y classification with the maximum P value is the predicted classification for the testing data.**

Different Naïve Bayes classifiers differ mainly by the assumption they make regarding the distribution of $P(y|x_1,...x_n)$.

Please see the following link for example of using NB algorithm for machine learning: http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/

### 3. Gaussian Naïve Bayes (Gaussian NB)

Gaussian NB implements the Gaussian Naïve Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

### 4. Multinomial Naive Bayes (Multinomial NB)

MultinomialNB implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in **text classification** where the data are typically represented as word vector counts (tf-idf vectors is another popular method for text classification).

$$P(x_i|y) = \frac{N_{yi}+\alpha}{N_y+\alpha n}$$

where $N_{yi} = \Sigma_{x\in T} x_i$ is the number of times features i appear in a sample of class y in the training set T, and $N_y = \Sigma_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha=1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

## 5. Bernoulli Naive Bayes (BernoulliNB)

BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions. Bernoulli distribution is the probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability 1-p.

$P(x_i \mid y)$ is defined as

$$P(x_i \mid y) = P(i \mid y) x_i + (1 - P(i \mid y))(1 - x_i)$$

## 6. Pros and Cons of Naïve Bayes Algorithm

Naive Bayes is an eager and fast learning classifier. When assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression and you need less training data.

However, it is known as a bad estimator, so the probability outputs from *predict_proba* are not to be taken too seriously. Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

**Reference:**


1, http://scikit-learn.org/stable/modules/naive_bayes.html

2, https://machinelearningmastery.com/naive-bayes-for-machine-learning/

3, http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/