Understanding the mathematics behind Naive Bayes

Naive Bayes, or called Naive Bayes classifier, is a classifier based on Bayes Theorem with the naive assumption that features are independent of each other. Without further ado, let's get straight to the derivation of the model.

Bayes Theorem

Given a feature vector $X = (x_1, x_2, \ldots, x_n)$ and a class variable C_k, Bayes Theorem states that:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}, \quad for\ k = 1, 2, \ldots, K$$

We call $P(C_k \mid X)$ the posterior probability, $P(X \mid C_k)$ the likelihood, $P(C_k)$ the prior probability of class, and $P(X)$ the prior probability of predictor. We are interested in calculating the posterior probability from the likelihood and prior probabilities.

Using the chain rule, the likelihood $P(X \mid C_k)$ can be decomposed as:

$$P(X \mid C_k) = P(x_1, \ldots, x_n \mid C_k) = P(x_1 \mid x_2, \ldots x_n, C_k)P(x_2 \mid x_3, \ldots x_n, C_k) \cdots P(x_{n-1} \mid x_n, C_k)P(x_n \mid C_k)$$

Naive independence assumption

The above sets of probabilities can be hard and expensive to calculate. Fortunately, with the naive conditional independence assumption, which is stated as:

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

We can get:

$$P(X \mid C_k) = P(x_1, \ldots, x_n \mid C_k) = \prod_{i=1}^{n} P(x_i \mid C_k)$$

And the posterior probability can then be written as:

$$P(C_k|X) = \frac{P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)}{P(X)}$$

Naive Bayes model

Since the prior probability of predictor $P(X)$ is constant given the input, we can get:

$$P(C_k|X) \propto P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

where $\propto$ means positive proportional to.

The Naive Bayes classification problem then becomes: for different class values of C_k, find the

maximum of $P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$. This can be formulated as:

$$\hat{C} = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

The prior probability of class $P(C_k)$ could be calculated as the relative frequency of class C_k in the training data.

Different Naive Bayes classifiers

Practically speaking, the likelihood $P(x_i \mid C_k)$ are usually modeled using the same class of probability distribution. The different Naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid C_k)$.

Some commonly used Naive Bayes classifiers include *Gaussian Naive Bayes* classifier, *Multinomial Naive Bayes* classifier, and *Bernoulli Naive Bayes* classifier.

Multinomial Naive Bayes and Bernoulli Naive Bayes are two classic naive Bayes classifiers used in text classification. It's hard to tell which one performs better than the other in general. It's best to run and evaluate both models.

Extra words about Naive Bayes

Remember we have a naive independence assumption about the features of the data. While this may seem overly simplistic or naive, in practice naive Bayes classifiers have worked quite well in many real-world applications.

Because of the independence assumption, naive Bayes classifiers can quickly learn to use high dimensional features with limited training data compared to more sophisticated methods. This can be useful in situations where the dataset is small compared to the number of features, such as images or texts.

The computational efficiency of Naive Bayes lies in the fact that the runtime complexity of Naive Bayes classifier is $O(nK)$, where n is the number of features and K is the number of label classes. This is especially useful in dealing with very high dimensional data, such as a large-corpus text dataset, or a high-resolution image dataset.

Explanation = https://youtu.be/2PVRG45eVrY?si=13_YkSdRjh9ozsf-