









☆ Key Points

Rule.



First you need to calculate the

probabilities of the labels given

the observed data, using Bayes'

Then, you choose y such that this

considering each dimension of the

feature vector independently.

probability is maximized,







- Course Shortcuts
- > Student Lounge
- > Q&A
- Estimating Distributions from Data
- Classification with the NaiveBayes Algorithm
 - Module Introduction:
 Classification with the
 Naive Bayes Algorithm
 - Limitations of High
 Dimensional Space in MLE
 - MLE Curse of Dimensionality
 - Formalize the Curse of Dimensionality
 - Capturing All Possibilities in d Dimensions
 - Naive Bayes Assumption
 - Data Sets Where the Naive Bayes Assumption Holds
 - Naive Bayes Classifier
 - Derivation of Naive Bayes Classifier
 - Naive Bayes Cheat Sheet
 - Determine Probability
 With Categorical Naive
 Bayes
 - <u>Categorical Naive Bayes</u> <u>Classifiers</u>
 - Sh Naive Bayes in Action
 - Module Wrap-up:
 Classification with the
 Naive Bayes Algorithm
- > Building a Baby NameClassifying System
- > Course Resources

Estimating Probability Distributions >

Derivation of Naive Bayes Classifier

The Naive Bayes classifier decomposes a single high dimensional probability estimation problem into many 1-dimensional estimation problems by assuming the features are conditionally independent given the class label. We want to estimate $\hat{P}(y|\mathbf{x})$ with the Naive Bayes classifier. To do so, we will make use of **Bayes' Rule and the Naive Bayes assumption**.

Breaking Down Naive Bayes

You can estimate $\hat{\mathbf{P}}(y|\mathbf{x})$ by calculating $\mathbf{P}(y)$ and $\mathbf{P}(\mathbf{x}|y)$, since by Bayes' Rule:

$$\mathrm{P}(y|\mathbf{x}) = rac{\mathrm{P}(\mathbf{x}|y)\mathrm{P}(y)}{\mathrm{P}(\mathbf{x})}$$

Estimating ${
m P}(y=c)$ (the probability that y takes on some value c) is simple - it is just the fraction of data points where y=c .

We can write this formally using an indicator variable I, which is 1 when the argument holds and 0 when it doesn't. If y takes on discrete binary values, such as 0 or 1, it simply counts how many times we observe each outcome:

$$\mathrm{P}(y=c) = rac{\sum_{i=1}^n I(y_i=c)}{n} = \hat{\pi}_c$$

Estimating $P(\mathbf{x}|y)$, however, is not so simple. To do so, you will make use of the Naive Bayes assumption from earlier:

$$\mathrm{P}(\mathbf{x}|y) = \prod_{lpha=1}^d \mathrm{P}(x_lpha|y)$$

where $x_{\alpha} = [\mathbf{x}]_{\alpha}$ is the value for feature α .

Below, you'll see an illustration of how the Naive Bayes algorithm decomposes a multi-dimensional probability estimation problem (first image) into many 1-dimensional estimation problems. First, you estimate $P(x_{\alpha}|y)$ independently in each dimension (center two images) and then obtain an estimate of the full data distribution by assuming conditional independence $P(\mathbf{x}|y) = \prod_{\alpha} P(x_{\alpha}|y)$ (rightmost image).

Original	Estimation of first dimension	Estimation of second dimension	Resulting data distribution
y = 1 $y = 2$	$P(\mathbf{x}_1 y=2)$ $P(\mathbf{x}_1 y=1)$	$P(\mathbf{x}_2 y=1) \qquad P(\mathbf{x}_2 y=2)$	$\prod_{\alpha} P(\mathbf{x}_{\alpha} y=2)$ $\prod_{\alpha} P(\mathbf{x}_{\alpha} y=1)$

Derivation of Naive Bayes Classifier

Now, we can begin our derivation of the Naive Bayes classifier as follows:

$$\begin{split} h(\mathbf{x}) &= \arg\max_{y} \frac{\mathrm{P}(y|\mathbf{x})}{\mathrm{P}(\mathbf{x})} \\ &= \arg\max_{y} \frac{\mathrm{P}(\mathbf{x}|y)\mathrm{P}(y)}{\mathrm{P}(\mathbf{x})} \\ &= \arg\max_{y} \mathrm{P}(\mathbf{x}|y)\mathrm{P}(y) \\ &= \arg\max_{y} \prod_{\alpha=1}^{d} \mathrm{P}(x_{\alpha}|y)\mathrm{P}(y) \\ &= \arg\max_{y} \sum_{\alpha=1}^{d} \log(\mathrm{P}(x_{\alpha}|y)) + \log(\mathrm{P}(y)) \end{split} \qquad \text{(as log is a monotonic function)}$$

Estimating $P(x_{\alpha}|y)$ is easy since we only need to consider one single dimension, so MLE should work well. Estimating P(y) is not affected by the assumption and can similarly done with straight-forward MLE.