The Naïve Bayes (NB) Classifier

- For two events A and B, Bayes’ theorem tells us the following:
  \[ P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \]
- In other words, if an event B occurs, and we know how likely that event is to occur and how likely it occurs if event A happens at the same time, then we can update our knowledge on the probability of A occurring.
- We can update the prior probability P(A) to the posterior probability P(A|B) after event B took place.

During training, we determine the probability distributions of all features for each of the classes.

- When classifying an input, we keep track of the probabilities for this input to belong to each of the classes.
- Typically, before analyzing the input vector, it is equally likely to belong to each class.
- Then we look at the value of the first feature and update the probability for each class according to Bayes’ rule and the probability distributions obtained during training.

The Artificial Neural Network (ANN) Classifier

- NNs are able to learn by adapting their connectivity patterns so that the organism improves its behavior in terms of reaching certain (evolutionary) goals.
- The strength of a connection, or whether it is excitatory or inhibitory, depends on the state of a receiving neuron’s synapses.
- The NN achieves learning by appropriately adapting the states of its synapses.

Supervised Function Approximation

In supervised learning, we train an artificial NN (ANN) with a set of vector pairs, so-called exemplars. Each pair \((x, y)\) consists of an input vector \(x\) and a corresponding output vector \(y\). Whenever the network receives input \(x\), we would like it to provide output \(y\).

The exemplars thus describe the function that we want to “teach” our network. Besides learning the exemplars, we would like our network to generalize, that is, give plausible output for inputs that the network had not been trained with.
Linear Separability

\[ f(x_1, x_2, ..., x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0, & \text{otherwise} \end{cases} \]

Input space in the two-dimensional case (n = 2):

So by varying the weights and the threshold, we can realize **any linear separation** of the input space into a region that yields output 1, and another region that yields output 0.

As we have seen, a **two-dimensional** input space can be divided by any straight line.

A **three-dimensional** input space can be divided by any two-dimensional plane.

In general, an **n-dimensional** input space can be divided by an (n-1)-dimensional plane or hyperplane.

Of course, for n > 3 this is hard to visualize.