The Bayes Classifier

We have been starting to look at the supervised classification problem: we are given data $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, and $y_i \in \{1, \ldots, K\}$. In this section, we suppose that we know everything there is to know about the data (in a probabilistic sense): we assume that we know the *joint distribution* of (X, Y). If we have full knowledge of the distribution, then we can design an optimal classifier without seeing any data at all.

We now make the mathematical setup completely concrete. The "feature vector" X is a random vector[1] in $\mathbb{R}^d$, and the "class label" Y is a discrete random scalar in $\{1, \ldots, K\}$. When we say that we have a joint probability distribution for (X, Y), it means that we have a rule that assigns probabilities to events that obeys the Kolmogorov axioms[2]. Given $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \{1, \ldots, K\}$, the joint distribution gives us a probability

$$\mathrm{P}\left[X \in \mathcal{X}, Y \in \mathcal{Y}\right] = \text{ probability that } X \text{ is in } \mathcal{X} \text{ and } Y \text{ is in } \mathcal{Y}.$$

We will treat the entries in the feature vector as continuous-valued. Fixing the feature vector X at different points $\boldsymbol{x}$ results in different conditional probability mass functions (pmfs) for the class label Y:

$$\eta_k(\boldsymbol{x}) := p_{Y|X}(k|\boldsymbol{x}) = \mathrm{P}\left[Y = k | X = \boldsymbol{x}\right]. \tag{1}$$

The pmf $p_{Y|X}(k|\boldsymbol{x})$, which is also called the **a posteriori distribution**, will play a central role in much of our discussion here and throughout the course. We encounter it so often that it is useful to give it the more compact notation $\eta_k(\boldsymbol{x})$.

[1] We use non-bold capital letters for all random variables in these notes, whether they are scalar-, vector-, matrix-, or whatever-valued.

[2] https://en.wikipedia.org/wiki/Probability_axioms

It is also useful to note that fixing the class label Y to different values y results in different conditional probability density functions (pdfs) for the feature vector X: $f_{X|Y}(\boldsymbol{x}|y)$, where

$$\mathrm{P}\left[X \in \mathcal{X} | Y = y\right] = \int_{\mathcal{X}} f_{X|Y}(\boldsymbol{x}|y)\, d\boldsymbol{x}.$$

$f_{X|Y}(\boldsymbol{x}|y)$ is the **class conditional distribution** of X, i.e., the distribution of X given that Y belongs to class y.

A classification rule or **classifier** is simply a function $h : \mathbb{R}^d \to \{1, \ldots, K\}$; that is, a function which takes a feature vector and returns a class label. We can specify this classification rule by **paritioning** $\mathbb{R}^d$ into K regions $\Gamma_1(h), \ldots, \Gamma_K(h)$, where $\Gamma_k(h)$ is the set of point that h maps to k:

$$\Gamma_k(h) = \{\boldsymbol{x} \in \mathbb{R}^d \ : \ h(\boldsymbol{x}) = k\}.$$

We will judge the quality of a classifier by the probability that it makes a mistake:

$$R(h) = \mathrm{P}\left[h(X) \neq Y\right].$$

This is also called the **risk** of h, or the **probability of error**.

We can now ask a very well-defined question which has a clear-cut answer: What is the classifier that minimizes the probability of error? The answer is simple: given $X = \boldsymbol{x}$, choose the class label that maximizes the conditional probability in (1).

Theorem: Define the classifier

$$h^*(\boldsymbol{x}) = \arg \max_{k \in \{1, \ldots, K\}} \eta_k(\boldsymbol{x}). \tag{2}$$

Then every other classifier h has

$$R(h) \geq R(h^*).$$

Proof: The optimality of h^* in (2) follows from carefully writing down the risk for an arbitrary classifier h, applying Bayes rule, and then showing that h^* optimizes the resulting expression. We start with an expression for $1 - R(h)$, which we will show is as *large* as possible when $h = h^*$:

$$\begin{aligned}
1 - R(h) &= \mathrm{P}\left[h(X) = Y\right] \\
&= \sum_{k=1}^{K} \mathrm{P}\left[Y = k\right] \cdot \mathrm{P}\left[h(X) = k | Y = k\right] \\
&= \sum_{k=1}^{K} \mathrm{P}\left[Y = k\right] \int_{\Gamma_k(h)} f_{X|Y}(\boldsymbol{x}|k)\, d\boldsymbol{x} \\
&= \sum_{k=1}^{K} \int_{\Gamma_k(h)} \mathrm{P}\left[Y = k\right] f_{X|Y}(\boldsymbol{x}|k)\, d\boldsymbol{x}.
\end{aligned}$$

By Bayes rule,

$$\eta_k(\boldsymbol{x}) = \frac{\mathrm{P}\left[Y = k\right] f_{X|Y}(\boldsymbol{x}|k)}{\sum_{\ell=1}^{K} \mathrm{P}\left[Y = \ell\right] f_{X|Y}(\boldsymbol{x}|\ell)}.$$

Note that the denominator is a function of $\boldsymbol{x}$ that is independent of k; it is in fact the marginal density $f_X(\boldsymbol{x})$ for X. Using this and the fact that the regions $\Gamma_k(h)$ are disjoint, we can continue the string of equalities:

$$1 - R(h) = \int_{\mathbb{R}^d} \left(\sum_{k=1}^{K} 1_{\Gamma_k(h)}(\boldsymbol{x})\, f_X(\boldsymbol{x}) \eta_k(\boldsymbol{x}) \right) d\boldsymbol{x},$$

where $1_{\mathcal{A}}(\boldsymbol{x})$ is the indicator function

$$1_{\mathcal{A}}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \mathcal{A}, \\ 0, & \boldsymbol{x} \notin \mathcal{A}. \end{cases}$$

The way we choose h^* in (2) chooses the regions so that the function inside the integral above is as large as possible; it is clear that

$$\sum_{k=1}^{K} 1_{\Gamma_k(h)}(\boldsymbol{x}) f_X(\boldsymbol{x}) \eta_k(\boldsymbol{x}) \;\leq\; \sum_{k=1}^{K} 1_{\Gamma_k(h^*)}(\boldsymbol{x}) f_X(\boldsymbol{x}) \eta_k(\boldsymbol{x}),$$

for all $\boldsymbol{x} \in \mathbb{R}^d$. Thus

$$\begin{aligned} 1 - R(h) &\leq \int_{\mathbb{R}^d} \left(\sum_{k=1}^{K} 1_{\Gamma_k(h)}(\boldsymbol{x})\, f_X(\boldsymbol{x}) \eta_k(\boldsymbol{x}) \right) \, d\boldsymbol{x} \\ &= 1 - R(h^*), \end{aligned}$$

and so $R(h^*) \leq R(h)$.

The nearest neighbor classifier

We have just seen that the Bayes classifier is optimal. Unfortunately, it requires complete knowledge of the conditional probability mass function $\eta_k(\boldsymbol{x})$. In the context of machine learning, this is not a reasonable assumption. The **nearest neighbor classifier** is an *extremely* simple alternative. For any $\boldsymbol{x}$, we simply find the closest point $\boldsymbol{x}_i$ in the training set and then assign $\boldsymbol{x}$ the same label as its nearest neighbor.

This is an incredibly simple rule, but perhaps somewhat surprisingly we can show that as $n \to \infty$, i.e., as the size of our training data grows, this simple classifier is near-optimal. To see this, we will consider the risk of the nearest neighbor classifier h^{NN} conditioned on $X = \boldsymbol{x}$ and compare this to the risk of the Bayes classifier h^*.

To make our discussion simpler, we will restrict our attention to the case of binary classification where $y_i \in \{0, 1\}$. We first note that the risk of the Bayes classifier h^* conditioned on $X = \boldsymbol{x}$ is given by

$$R^*(\boldsymbol{x}) := \mathrm{P}\left[Y \neq h^*(\boldsymbol{x}) | X = \boldsymbol{x}\right].$$

If $h^*(\boldsymbol{x}) = 0$ then we have $R^*(\boldsymbol{x}) = \mathrm{P}\left[Y = 1 | X = \boldsymbol{x}\right] = \eta_1(\boldsymbol{x})$. Similarly, if $h^*(\boldsymbol{x}) = 1$ we have $R^*(\boldsymbol{x}) = \eta_1(\boldsymbol{x})$. Since by definition $h^*(\boldsymbol{x})$ selects the label that *maximizes* $\eta_k(\boldsymbol{x})$, we thus have that

$$R^*(\boldsymbol{x}) = \min\{\eta_0(\boldsymbol{x}), \eta_1(\boldsymbol{x})\}. \tag{3}$$

For the nearest neighbor classifier, note that

$$R^{\mathrm{NN}}(\boldsymbol{x}) := \mathrm{P}\left[h^{\mathrm{NN}}(\boldsymbol{x}) \neq Y | X = \boldsymbol{x}\right].$$

In our analysis, we will treat not only (X, Y) as random, but also the output $h^{\mathrm{NN}}(\boldsymbol{x})$ as random since it depends on the dataset, which is itself drawn at random from the same distribution as (X, Y). This allows us to write

$$\begin{aligned} R^{\mathrm{NN}}(\boldsymbol{x}) = & \, \mathrm{P}\left[Y = 0 | X = \boldsymbol{x}\right] \mathrm{P}\left[h^{\mathrm{NN}}(\boldsymbol{x}) = 1 | X = \boldsymbol{x}\right] \\ & + \mathrm{P}\left[Y = 1 | X = \boldsymbol{x}\right] \mathrm{P}\left[h^{\mathrm{NN}}(\boldsymbol{x}) = 0 | X = \boldsymbol{x}\right]. \end{aligned} \tag{4}$$

If $\boldsymbol{x}_{\mathrm{NN}}$ denotes the nearest neighbor to $\boldsymbol{x}$, then we can write

$$\mathrm{P}\left[h^{\mathrm{NN}}(\boldsymbol{x}) = k | X = \boldsymbol{x}\right] = \mathrm{P}\left[Y = k | X = \boldsymbol{x}_{\mathrm{NN}}\right] = \eta_k(\boldsymbol{x}_{\mathrm{NN}}).$$

As $n \to \infty$, we have that $\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}\| \to 0$, and thus as $n \to \infty$ we have

$$\eta_k(\boldsymbol{x}_{\mathrm{NN}}) \to \eta_k(\boldsymbol{x}).$$

Plugging this back into (4) and simplifying, we obtain

$$\begin{aligned} R^{\mathrm{NN}}(\boldsymbol{x}) & \to \eta_0(\boldsymbol{x})\eta_1(\boldsymbol{x}) + \eta_1(\boldsymbol{x})\eta_0(\boldsymbol{x}) \\ & = 2\eta_0(\boldsymbol{x})\eta_1(\boldsymbol{x}) \\ & \leq 2\min\{\eta_0(\boldsymbol{x}), \eta_1(\boldsymbol{x})\}, \end{aligned}$$

where the last inequality follows from the fact that both $\eta_1(\boldsymbol{x})$ and $\eta_2(\boldsymbol{x})$ are less than 1. Combining this with (3), this yields

$$\lim_{n\to\infty} R^{\mathrm{NN}}(\boldsymbol{x}) \leq 2R^*(\boldsymbol{x}),$$

or in words, that asymptotically, the risk of the nearest neighbor classifier is at most twice the Bayes risk.

This can be strengthened by considering the more general k-nearest neighbors classifier. The idea here is to assign a label to $\boldsymbol{x}$ by taking a majority vote over the k training points closest to $\boldsymbol{x}$. If $R^{\mathrm{kNN}}(\boldsymbol{x})$ denotes the risk of the k-nearest neighbor classifier, then one can show via a similar argument that

$$\lim_{n\to\infty} R^{\mathrm{kNN}}(\boldsymbol{x}) \leq \left(1 + \sqrt{2/k}\right) R^*(\boldsymbol{x}).$$

Thus, by increasing k it is possible to drive this multiplicative constant arbitrarily close to 1. This results in a property known as **universal consistency**. Specifically, if R^* denotes the Bayes risk and R_n^{kNN} denotes the risk of the k-nearest neighbors classifier based on a dataset of size n, then one can show that as $n \to \infty$, if $k \to \infty$ while $k/n \to 0$, then $R_n^{\mathrm{kNN}} \to R^*$.

In words this is simply saying that for any possible distribution on the data, if we are given enough data eventually the risk of the k-nearest neighbor classifier will converge to the Bayes risk (i.e., to the optimal risk). Unfortunately (or fortunately, depending on your perspective), you might have to wait a very long time, so there is still a role for other machine learning algorithms to improve on this situation when we only have a finite amount of data.