4

The Bias-Variance Tradeoff

Given a fixed hypothesis set $\mathcal{H}$ consisting of binary classifiers $h\colon \mathcal{X} \to \mathcal{Y} = \{0, 1\}$, we would like to find a classifier $\hat{h}_n \in \mathcal{H}$, computed from n i.i.d. random samples (X_i, Y_i), that has a small **excess risk**

$$\mathcal{E}(\hat{h}_n) = R(\hat{h}_n) - R(h^*), \tag{A}$$

where h^* denotes the Bayes classifiers and $R(h) = \mathbb{P}(h(X) \neq Y)$. The excess risk may be large because the best achievable risk among classifiers in $\mathcal{H}$ is far from $R(h^*)$. But even if $\mathcal{H}$ is expressive enough to contain h^*, finding the best possible classifier in $\mathcal{H}$ may not be feasible. We will mostly be concerned with the case when $\hat{h}_n$ is constructed by minimizing the empirical risk over the class $\mathcal{H}$:

$$\hat{R}(\hat{h}_n) = \inf_{h \in \mathcal{H}} \hat{R}(h), \quad \text{where} \quad \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{h(X_i) \neq Y_i\}.$$

As $\hat{h}_n$ is constructed by minimizing the average over a finite sample set, rather than the expected value over the whole space, the resulting $\hat{h}_n$ will usually not minimize $R(h)$ over $\mathcal{H}$. Less ambitious than trying to minimize the excess risk is therefore the goal of bounding the difference in risk between $\hat{h}_n$ and the best possible $h \in \mathcal{H}$. In this chapter we study the relationship between the error due to the limitations of the class $\mathcal{H}$ to the error due to the limitations of working with finite data samples.

Approximation and Estimation

Given a set of candidate classifiers $\mathcal{H}$, we can decompose the generalization risk $R(\hat{h}_n)$ as follows:

$$R(\hat{h}_n) = \underbrace{R(\hat{h}_n) - \inf_{h \in \mathcal{H}} R(h)}_{\text{Estimation error}} + \underbrace{\inf_{h \in \mathcal{H}} R(h) - R(h^*)}_{\text{Approximation error}} + \underbrace{R(h^*)}_{\text{Irreducible error}}$$

The first part compares the performance of $\hat{h}_n$ against the best possible classifier *within the class* $\mathcal{H}$, while the second part is a statement about the power of the class $\mathcal{H}$ itself: it tells us how close we can get to the Bayes classifier if we stay within $\mathcal{H}$. We can reduce the estimation error by making the class $\mathcal{H}$ smaller, but then the approximation error increases. Note that here we consider $\hat{h}_n$, and hence $R(\hat{h}_n)$, as a random variable, constructed from n i.i.d. random samples $\{(X_i, Y_i)\}_{i=1}^n$.

Example 4.1. There is usually no unique minimizer of $\hat{R}(h)$ within a class $\mathcal{H}$. Figure 4.1 shows data generated by taking $X = (X_1, X_2)$ to be uniformly distributed on a square in $\mathbb{R}^2$, and choosing Y from a Bernoulli distribution with probability p that is $1/2$ the line $x_2 = 4 - x_1$, increases to 1 with the distance to this line if $X_2 > 4 - X_1$, and decreases to 0 with the distance if $X_2 < 4 - X_1$. Four classifiers $\hat{h}_n$ are chosen from standard sets of classifiers in machine learning (we will study Support Vector Machines (SVM) in more detail later). On the training data, the accuracy (percentage of correctly classified points) of the four methods is 92.5%, 95%, 97.5% and 100%, respectively. On the whole distribution, though, the linear SVM classifier performs best, with an accuracy of 87.7%, while the decision tree classifier, that fitted the training data perfectly, only achieves 82%.

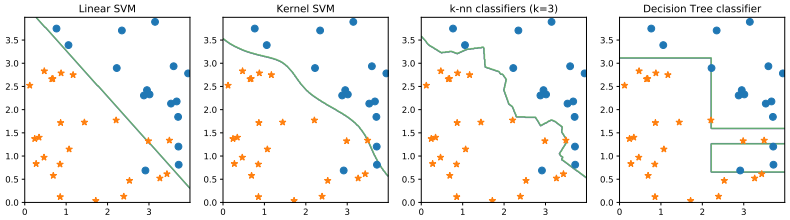

Figure 4.1: For decision boundaries for some standard classifiers.

In this particular example, the Bayes classifier itself is a linear classifier (we assign values $y = 1$ or $y = 0$ to an input x depending on whether that point lies on one side of a straight line or another), and hence the class $\mathcal{H}$ of linear classifiers has approximation error 0. This is not necessarily true for other, possibly more extensive classes of functions that can have a smaller estimation error. A classifier is a deterministic rule: given an input $\boldsymbol{x}$, it assigns to it a label y. Adapting a classifier too closely to observed training data is called **overfitting**, and is generally considered to be the worst possible crime in data science. Informally, one can think of overfitting as trying to model the noise, and not just the relationships in the data.

Example 4.2. We now consider again points in $\mathbb{R}^2$ that are split in two groups, and consider the class $\mathcal{H}_n$ of k-nearest neighbour classifiers. Given a reference data

set $\{(\boldsymbol{x}^i, y^i)\}_i = 1^n$, a k-th nearest neighbour classifier $\hat{h}_n$ is constructed by first computing, for each $\boldsymbol{x}$, the average

$$\frac{1}{k} \sum_{i \in N_k(\boldsymbol{x})} y^i,$$

where $N_k(\boldsymbol{x})$ consists of those indices $j \in \{1, \ldots, n\}$ that correspond to the closest points to $\boldsymbol{x}$ among the $\boldsymbol{x}^j$. We then assign the label $y = 1$ if this average is greater than $1/2$ and 0 if it is smaller than $1/2$ (hence, the label is assigned based on the label of the majority of the k closest neighbours). The distance between points can be measured in various ways; here, we use the usual Euclidean distance on the plane. Figure 4.2 show the decision boundaries for k-nearest classifiers for various k.

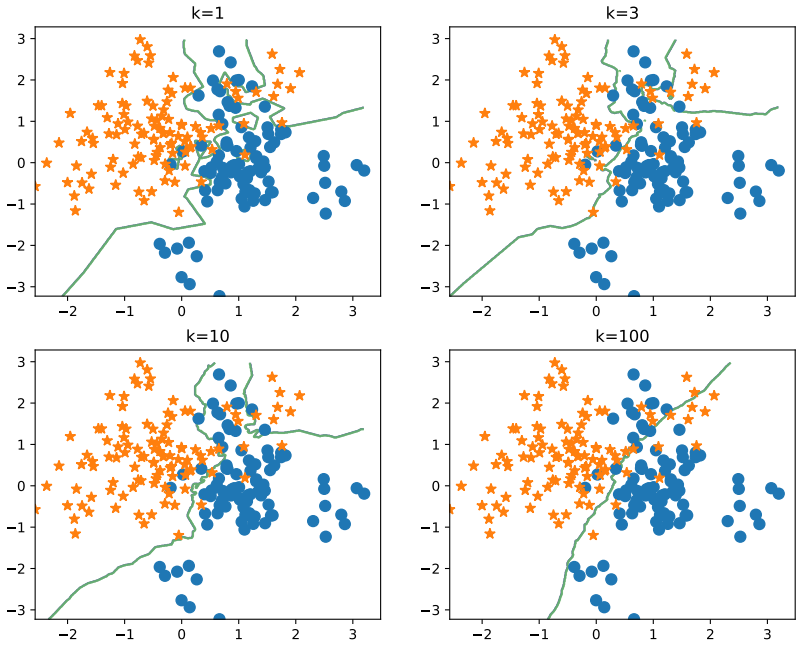

Figure 4.2: The decision boundary of four k-nearest neighbour classifiers.

How do we evaluate which of these performs best, in the sense of having the smallest generalization risk? Since we do not (or pretend to not) have access to the distribution that generated the data, the only sensible option is to sample more data and evaluate compute the empirical error on this additional data. Such additional data is called **test data**. In machine learning, one typically splits the available data randomly into training and test data, uses the training data to construct the classifier, and the test data to evaluate the classifier by computing an approximation to the generalization risk.

Figure 4.3 compares the classification error on the training set with the generalization risk (computed by generating a very large number of additional samples) and the Bayes risk (which we can compute here, since we know the distribution from which the data was generated). In this example, the data consists of a mixture of two mixtures of 10 Gaussians each.

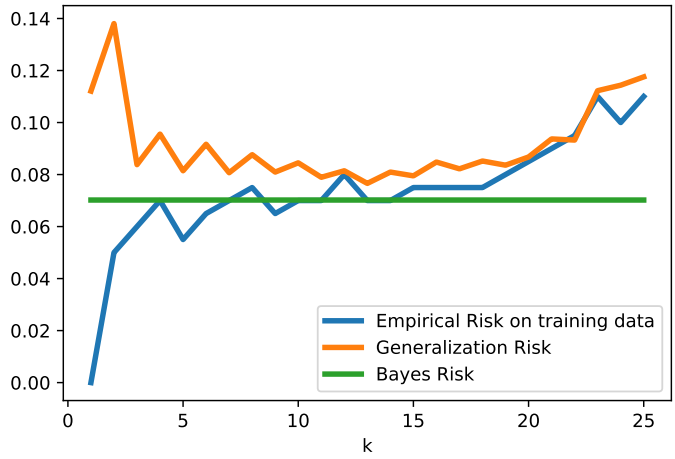

Figure 4.3: The empirical risk on training data compared to the generalization risk and the Bayes risk for k-nearest neighbour classifiers and different values of k. The different between the upper curve and the Bayes error at each k is the approximation error for the set of k-nearest neighbour classifiers.

We see that the 1-nearest neighbour classifier does best on the training data (100% accuracy) but becomes less accurate on the training data as k increases. On the other hand, the generalization risk appears to be optimal at around a value of $k = 13$. The parameter k is related to the **variance** of a classifier. For small k the variance is high, in the sense that the resulting classifier is closely adapted ("overfitted") to a particular set of samples, while for large k the variance is small, in the sense that the decision boundary will not change much with changes in the training data (see, for example, the difference between the $k = 1$ and the $k = 100$ displays in Figure 4.2).

The Bias-Variance Tradeoff

The dichotomy between estimation and approximation is closely related to the concept of **bias-variance** tradeoff in statistics. To see this connection, note that we can express the generalization risk as means squared error:

$$R(\hat{h}_n) = \mathbb{P}(\hat{h}_n(X) \neq Y) = \mathbb{E}[(\hat{h}_n(X) - Y)^2].$$

Recall that the Bayes classifier was defined in terms of the regression function $f(X) = \mathbb{E}[Y|X]$, and that the response variable Y could be characterized as

$$Y = f(X) + \epsilon,$$

where ϵ can be interpreted as random noise with $\mathbb{E}[\epsilon|X] = 0$, while f describes the relationship between input and output data that we would like to capture. The variance

$$\sigma^2 = \mathbb{E}[\epsilon^2] = \mathbb{E}[(Y - f(X))^2]$$

is called the **irreducible error**. A positive irreducible error prevents the Bayes risk from being zero. The well-known bias-variance decomposition in statistics, applied to our context, can be formulated as

$$\begin{aligned}\mathbb{E}[(Y - \hat{h}_n(X))^2] &= \mathbb{E}[(Y - \mathbb{E}[\hat{h}_n])^2] + \mathbb{E}[(\hat{h}_n(X) - \mathbb{E}[\hat{h}_n])^2] \\ &= \mathbb{E}[\underbrace{(f(X) - \mathbb{E}[\hat{h}_n])}_{\text{bias}}{}^2] + \underbrace{\mathbb{E}[(\hat{h}_n(X) - \mathbb{E}[\hat{h}_n])^2]}_{\text{variance}} + \underbrace{\sigma^2}_{\text{irreducible error}}.\end{aligned}$$

The expectation here is taken over both (X, Y) and random training data $\{(X_i, Y_i)\}_{i=1}^n$.

Bounding the estimation error

In this chapter we focus on the estimation error and leave the problem of choosing an adequate class $\mathcal{H}$ for later. Assume there exists a best possible classifier $\overline{h} \in \mathcal{H}$,

$$\overline{h} \in \arg\min_{h \in \mathcal{H}} R(h).$$

The notation emphasizes that such an optimal classifier does not have to be unique. The classifier $\overline{h}$ depends only on the class $\mathcal{H}$ and the probability distribution. Since $R(\hat{h}_n)$ is a random variable (as mentioned above, it depends on the n i.i.d. samples (X_i, Y_i)), any bounds on the difference $R(\hat{h}_n) - R(\overline{h})$ are necessarily probabilistic. More precisely, for any given *tolerance* $\delta \in (0, 1)$, we want to find a bound $C(n, \delta)$ such that

$$R(\hat{h}_n) - R(\overline{h}) \leq C(n, \delta)$$

holds with probability $1 - \delta$. Ideally, the constants should also depend on properties of the set $\mathcal{H}$, for example the size of $\mathcal{H}$ if this set is finite. In addition, we would like the bound to decrease to 0 as $n \to \infty$.

If we denote by $\overline{h}$ the minimizer of $R(h)$ over $\mathcal{H}$, then we can decompose the estimation error as

$$\begin{aligned}R(\hat{h}_n) - R(\overline{h}) &= \overbrace{\hat{R}(\hat{h}_n) - \hat{R}(\overline{h})}^{\leq 0} + R(\hat{h}_n) - \hat{R}(\hat{h}_n) + \hat{R}(\overline{h}) - R(\overline{h}) \\ &\leq 2 \sup_{h \in \mathcal{H}} |R(h) - \hat{R}(h)|.\end{aligned} \tag{4.1}$$

The inequality $\hat{R}(\hat{h}_n) - \hat{R}(\overline{h}) \leq 0$ holds by definition, since $\hat{h}_n$ is the minimizer of the empirical risk. We thus arrive at the following problem: given $\delta \in (0,1)$, find a bound $C(n,\delta)$ such that

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} |R(h) - \hat{R}(h)| > C(n,\delta)\right) \leq \delta.$$

This problem brings us into the territory of the study of suprema of stochastic processes, and concentration of measure inequalities will play an important role in deriving such bounds.

Notes

See [2, Chapter 2] for a more detailed discussion of the bias-variance tradeoff in the case of linear regression and k-nearest neighbour classification. In particular, our Example 4.2 is taken from this source. The decomposition (4.1) was observed by Vapnik and Chervonenkis [3], see also [1, Chapter 8] for a comprehensive treatment.

[1] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[2] T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[3] V. N. Vapnik and A. Ya. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, USSR, 1974.