4.2.3 Example

We continue the example of polynomial regression to illustrate how the generalisation performance depends on the model complexity and the size of the training data.

Generalisation Performance and Model Complexity

Figure 4.5(a) shows the training and prediction loss of the fitted polynomial regression model $\hat{h}_\lambda$ as a function of the degree of the polynomial (model complexity) λ. We can see that the prediction loss and training loss are generally not the same, i.e.

$$\mathcal{J}(\hat{h}_\lambda) \neq J^*_\lambda. \tag{4.29}$$

In the figure, the prediction loss is smallest for $\lambda = 4$, and while a five-degree polynomial has the smallest training loss, it has the largest prediction loss. Such a mismatch between training and prediction performance is due to overfitting. The estimated model $\hat{h}_\lambda$ is highly tuned to the specific training data $\mathcal{D}^{\text{train}}$ and does not reflect the general relationship between the predictor and the target variable. In contrast, we see that increasing the complexity of the degree-zero or degree-one polynomial will decrease the prediction loss. That is, these models are underfitting the training data.

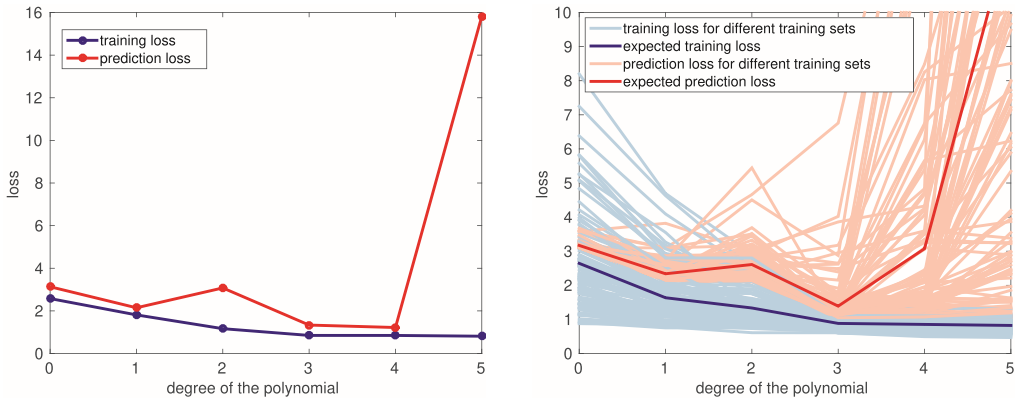
While Figure 4.5(a) depicts the training and prediction loss for a particular training set, Figure 4.5(b) shows their distribution over different training data sets. We can see that the variability of the prediction loss increases with the flexibility of the model. This is due to overfitting because the estimated model then depends strongly on the particularities of each training set that are bound to vary when the training data change. Underfitting, in contrast, leads to a small variability of the prediction loss because the fitted model captures comparably few properties of the training data.

The red solid line in Figure 4.5(b) shows the expected (average) prediction loss $\bar{\mathcal{J}}$ as a function of λ. While a model of degree $\lambda = 4$ performed best for the particular training data used in (a), models of degree $\lambda = 3$ yield the best performance on average. We see that there is here a difference between the generalisation performance of a specific fitted model and the generalisation performance of a model-family across different training sets, which reflects the general difference between $\mathcal{J}(\hat{h}_{\boldsymbol{\lambda}})$ and $\bar{\mathcal{J}}(\mathcal{A}_{\boldsymbol{\lambda}})$ discussed in Section 4.2.1.

Generalisation Performance and the Size of the Training Data

The results so far were obtained for training sets of size $n = 20$. We saw that flexible models tended to overfit the training data, so that there was stark difference between training and prediction performance. Here, we illustrate how the size of the training data influences the generalisation performance.

Figure 4.6 shows the expected training and prediction loss as a function of the size n of the training data for polynomial models of different degree. We can generally see that the training and prediction loss approach each other as the sample size increases. Note that they may generally not reach the same limit as n increases because the training and prediction loss functions L and $\mathcal{L}$, for example, may not be the same.



(a) Single training set

(b) Distribution over training sets

Figure 4.5: Training versus prediction performance of different prediction models.

Figure 4.6(a) shows that increasing the model complexity decreases the prediction loss for the models of degree zero and one. Moreover, their prediction loss does not decrease below a certain level even if the size of the training data increases. Both phenomena are a sign of underfitting.

Figure 4.6(b) shows the average training and prediction loss for the polynomial model of degree five. The large difference between training and prediction loss for small sample sizes is due to overfitting. As the size of the training data increases, however, the gap between the two losses becomes smaller, which means that the amount of overfitting decreases.

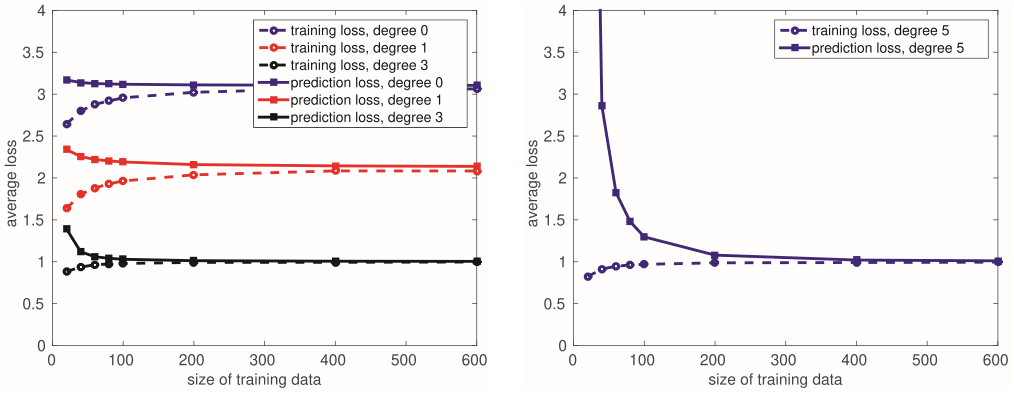
Comparing Figure 4.6(a) and (b) shows us further that even for large samples, on average, the model of degree five does here not achieve a smaller prediction loss than the model of degree three. Hence, for this problem, there is no advantage in using a more complex model than the model of degree three. In general, we can use model selection to choose among candidate models, or regularisation to avoid overfitting flexible models on small training data. Both model selection and choosing the right amount of regularisation correspond to hyperparameter selection.

4.3 Estimating the Generalisation Performance

We typically need to estimate the generalisation performance twice: Once for hyperparameter selection, and once for final performance evaluation. We first discuss two methods for estimating the generalisation performance and then apply them to the two aforementioned tasks.

4.3.1 Methods for Estimating the Generalisation Performance

The hold-out and the cross-validation approach to estimate the generalisation performance are presented.



(a) Polynomials of degree zero, one, and three

(b) Polynomial of degree five

Figure 4.6: Average training versus average prediction performance for different sizes of the training data.

Hold-out Approach

Assume that the prediction function $\hat{h}$ has been obtained using training data $\mathcal{D}^{\text{train}}$, i.e.

$$\hat{h} = \mathcal{A}(\mathcal{D}^{\text{train}}). \tag{4.30}$$

If another data set $\tilde{\mathcal{D}}$ is available with $\tilde{n}$ samples $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i) \sim p(\boldsymbol{x}, y)$ that are statistically independent from the samples in $\mathcal{D}^{\text{train}}$, we can use $\tilde{\mathcal{D}}$ to estimate the prediction loss $\mathcal{J}(\hat{h})$ via a sample average

$$\hat{\mathcal{J}}(\hat{h}; \tilde{\mathcal{D}}) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \mathcal{L}(\hat{h}(\tilde{\boldsymbol{x}}_i), \tilde{y}_i). \tag{4.31}$$

Depending on the context, $\tilde{\mathcal{D}}$ is called a test or a validation set.

We are typically given the union of the two data sets $\mathcal{D}^{\text{train}}$ and $\tilde{\mathcal{D}}$, and it is up to us how to split them into the two sets. Common split ratios are $n/\tilde{n} = 60/40$, $70/30$, or $80/20$. If the number of (hyper) parameters is large, it is better to increase the ratio so that more data are available for training.

While the splitting is often done randomly, particularly in classification, it is important that the different values of the target variable (e.g. the class labels) represented in a balanced way in both $\mathcal{D}^{\text{train}}$ and $\tilde{\mathcal{D}}$. Stratification methods can be used so that e.g. the classes are present in the same proportions in both $\mathcal{D}^{\text{train}}$ and $\tilde{\mathcal{D}}$.

The value of the estimated prediction loss in (4.31) may vary strongly for different hold-out data sets $\tilde{\mathcal{D}}$ unless $\tilde{n}$ is large. This is often seen as a drawback of the hold-out approach. Figure 4.7 illustrates the variability that can be introduced by randomly splitting a data set into a training set $\mathcal{D}^{\text{train}}$ and test set $\tilde{\mathcal{D}}$. Cross-validation is often used to avoid such issues.

Cross-validation

Cross-validation consists in randomly dividing the data that are available for training into K (roughly) equally-sized subset (folds) $\mathcal{D}_1, \ldots, \mathcal{D}_K$ without overlap.