CHAPTER 2

STATISTICAL LEARNING

The purpose of this chapter is to introduce the reader to some common concepts and themes in statistical learning. We discuss the difference between supervised and unsupervised learning, and how we can assess the predictive performance of supervised learning. We also examine the central role that the linear and Gaussian properties play in the modeling of data. We conclude with a section on Bayesian learning. The required probability and statistics background is given in Appendix C.

2.1 Introduction

Although structuring and visualizing data are important aspects of data science, the main challenge lies in the mathematical analysis of the data. When the goal is to interpret the model and quantify the uncertainty in the data, this analysis is usually referred to as *statistical learning*. In contrast, when the emphasis is on making predictions using large-scale data, then it is common to speak about *machine learning* or *data mining*.

STATISTICAL LEARNING

MACHINE LEARNING

DATA MINING

There are two major goals for modeling data: 1) to accurately predict some future quantity of interest, given some observed data, and 2) to discover unusual or interesting patterns in the data. To achieve these goals, one must rely on knowledge from three important pillars of the mathematical sciences.

Function approximation. Building a mathematical model for data usually means understanding how one data variable depends on another data variable. The most natural way to represent the relationship between variables is via a mathematical function or map. We usually assume that this mathematical function is not completely known, but can be approximated well given enough computing power and data. Thus, data scientists have to understand how best to approximate and represent functions using the least amount of computer processing and memory.

Optimization. Given a class of mathematical models, we wish to find the best possible model in that class. This requires some kind of efficient search or optimization procedure. The optimization step can be viewed as a process of fitting or calibrating a function to observed data. This step usually requires knowledge of optimization algorithms and efficient computer coding or programming.

Probability and Statistics. In general, the data used to fit the model is viewed as a realization of a random process or numerical vector, whose probability law determines the accuracy with which we can predict future observations. Thus, in order to quantify the uncertainty inherent in making predictions about the future, and the sources of error in the model, data scientists need a firm grasp of probability theory and statistical inference.

2.2 Supervised and Unsupervised Learning

FEATURE
RESPONSE

Given an input or *feature* vector $\boldsymbol{x}$, one of the main goals of machine learning is to predict an output or *response* variable y. For example, $\boldsymbol{x}$ could be a digitized signature and y a binary variable that indicates whether the signature is genuine or false. Another example is where $\boldsymbol{x}$ represents the weight and smoking habits of an expecting mother and y the birth weight of the baby. The data science attempt at this prediction is encoded in a mathematical function g, called the *prediction function*, which takes as an input $\boldsymbol{x}$ and outputs a guess $g(\boldsymbol{x})$ for y (denoted by $\widehat{y}$, for example). In a sense, g encompasses all the information about the relationship between the variables $\boldsymbol{x}$ and y, excluding the effects of chance and randomness in nature.

PREDICTION FUNCTION

REGRESSION

In *regression* problems, the response variable y can take any real value. In contrast, when y can only lie in a finite set, say $y \in \{0, \ldots, c-1\}$, then predicting y is conceptually the same as classifying the input $\boldsymbol{x}$ into one of c categories, and so prediction becomes a *classification* problem.

CLASSIFICATION

LOSS FUNCTION

We can measure the accuracy of a prediction $\widehat{y}$ with respect to a given response y by using some *loss function* $\text{Loss}(y, \widehat{y})$. In a regression setting the usual choice is the squared-error loss $(y-\widehat{y})^2$. In the case of classification, the zero–one (also written 0–1) loss function $\text{Loss}(y, \widehat{y}) = \mathbb{1}\{y \neq \widehat{y}\}$ is often used, which incurs a loss of 1 whenever the predicted class $\widehat{y}$ is not equal to the class y. Later on in this book, we will encounter various other useful loss functions, such as the cross-entropy and hinge loss functions (see, e.g., Chapter 7).

The word *error* is often used as a measure of distance between a "true" object y and some approximation $\widehat{y}$ thereof. If y is real-valued, the absolute error $|y-\widehat{y}|$ and the squared error $(y-\widehat{y})^2$ are both well-established error concepts, as are the norm $\|\boldsymbol{y}-\widehat{\boldsymbol{y}}\|$ and squared norm $\|\boldsymbol{y}-\widehat{\boldsymbol{y}}\|^2$ for vectors. The squared error $(y-\widehat{y})^2$ is just one example of a loss function.

It is unlikely that any mathematical function g will be able to make accurate predictions for all possible pairs $(\boldsymbol{x}, y)$ one may encounter in Nature. One reason for this is that, even with the same input $\boldsymbol{x}$, the output y may be different, depending on chance circumstances or randomness. For this reason, we adopt a probabilistic approach and assume that each pair $(\boldsymbol{x}, y)$ is the outcome of a random pair $(\boldsymbol{X}, Y)$ that has some joint probability density $f(\boldsymbol{x}, y)$. We then assess the predictive performance via the expected loss, usually called the *risk*, for g:

RISK

$$\ell(g) = \mathbb{E}\,\text{Loss}(Y, g(\boldsymbol{X})). \tag{2.1}$$

For example, in the classification case with zero–one loss function the risk is equal to the probability of incorrect classification: $\ell(g) = \mathbb{P}[Y \neq g(\boldsymbol{X})]$. In this context, the prediction

function g is called a *classifier*. Given the distribution of $(\boldsymbol{X}, Y)$ and any loss function, we can in principle find the best possible $g^* := \operatorname{argmin}_g \mathbb{E}\,\mathrm{Loss}(Y, g(\boldsymbol{X}))$ that yields the smallest risk $\ell^* := \ell(g^*)$. We will see in Chapter 7 that in the classification case with $y \in \{0, \ldots, c-1\}$ and $\ell(g) = \mathbb{P}[Y \neq g(\boldsymbol{X})]$, we have

CLASSIFIER

☞ 253

$$g^*(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \{0,\ldots,c-1\}} f(y \,|\, \boldsymbol{x}),$$

where $f(y \,|\, \boldsymbol{x}) = \mathbb{P}[Y = y \,|\, \boldsymbol{X} = \boldsymbol{x}]$ is the conditional probability of $Y = y$ given $\boldsymbol{X} = \boldsymbol{x}$. As already mentioned, for regression the most widely-used loss function is the squared-error loss. In this setting, the optimal prediction function g^* is often called the *regression function*. The following theorem specifies its exact form.

REGRESSION FUNCTION

Theorem 2.1: Optimal Prediction Function for Squared-Error Loss

For the squared-error loss $\mathrm{Loss}(y, \widehat{y}) = (y - \widehat{y})^2$, the optimal prediction function g^* is equal to the conditional expectation of Y given $\boldsymbol{X} = \boldsymbol{x}$:

$$g^*(\boldsymbol{x}) = \mathbb{E}[Y \,|\, \boldsymbol{X} = \boldsymbol{x}].$$

Proof: Let $g^*(\boldsymbol{x}) = \mathbb{E}[Y \,|\, \boldsymbol{X} = \boldsymbol{x}]$. For any function g, the squared-error risk satisfies

$$\begin{aligned}
\mathbb{E}(Y - g(\boldsymbol{X}))^2 &= \mathbb{E}[(Y - g^*(\boldsymbol{X}) + g^*(\boldsymbol{X}) - g(\boldsymbol{X}))^2] \\
&= \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}[(Y - g^*(\boldsymbol{X}))(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))] + \mathbb{E}(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))^2 \\
&\geqslant \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}[(Y - g^*(\boldsymbol{X}))(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))] \\
&= \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}\left\{(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))\mathbb{E}[Y - g^*(\boldsymbol{X}) \,|\, \boldsymbol{X}]\right\}.
\end{aligned}$$

In the last equation we used the tower property. By the definition of the conditional expectation, we have $\mathbb{E}[Y - g^*(\boldsymbol{X}) \,|\, \boldsymbol{X}] = 0$. It follows that $\mathbb{E}(Y - g(\boldsymbol{X}))^2 \geqslant \mathbb{E}(Y - g^*(\boldsymbol{X}))^2$, showing that g^* yields the smallest squared-error risk. □

☞ 433

One consequence of Theorem 2.1 is that, conditional on $\boldsymbol{X} = \boldsymbol{x}$, the (random) response Y can be written as

$$Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}), \tag{2.2}$$

where $\varepsilon(\boldsymbol{x})$ can be viewed as the random deviation of the response from its conditional mean at $\boldsymbol{x}$. This random deviation satisfies $\mathbb{E}\,\varepsilon(\boldsymbol{x}) = 0$. Further, the conditional variance of the response Y at $\boldsymbol{x}$ can be written as $\mathbb{V}\mathrm{ar}\,\varepsilon(\boldsymbol{x}) = v^2(\boldsymbol{x})$ for some unknown positive function v. Note that, in general, the probability distribution of $\varepsilon(\boldsymbol{x})$ is unspecified.

Since, the optimal prediction function g^* depends on the typically unknown joint distribution of $(\boldsymbol{X}, Y)$, it is not available in practice. Instead, all that we have available is a finite number of (usually) independent realizations from the joint density $f(\boldsymbol{x}, y)$. We denote this sample by $\mathcal{T} = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ and call it the *training set* ($\mathcal{T}$ is a mnemonic for training) with n examples. It will be important to distinguish between a random training set $\mathcal{T}$ and its (deterministic) outcome $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$. We will use the notation τ for the latter. We will also add the subscript n in τ_n when we wish to emphasize the size of the training set.

TRAINING SET

Our goal is thus to "learn" the unknown g^* using the n examples in the training set $\mathcal{T}$. Let us denote by $g_{\mathcal{T}}$ the best (by some criterion) approximation for g^* that we can construct

from $\mathcal{T}$. Note that $g_{\mathcal{T}}$ is a random function. A particular outcome is denoted by g_τ. It is often useful to think of a teacher–learner metaphor, whereby the function $g_{\mathcal{T}}$ is a *learner* who learns the unknown functional relationship $g^* : \boldsymbol{x} \mapsto y$ from the training data $\mathcal{T}$. We can imagine a "teacher" who provides n examples of the true relationship between the output Y_i and the input $\boldsymbol{X}_i$ for $i = 1, \ldots, n$, and thus "trains" the learner $g_{\mathcal{T}}$ to predict the output of a new input $\boldsymbol{X}$, for which the correct output Y is not provided by the teacher (is unknown).

LEARNER

SUPERVISED LEARNING

The above setting is called *supervised learning*, because one tries to learn the functional relationship between the feature vector $\boldsymbol{x}$ and response y in the presence of a teacher who provides n examples. It is common to speak of "explaining" or predicting y on the basis of $\boldsymbol{x}$, where $\boldsymbol{x}$ is a vector of *explanatory variables*.

EXPLANATORY VARIABLES

An example of supervised learning is email spam detection. The goal is to train the learner $g_{\mathcal{T}}$ to accurately predict whether any future email, as represented by the feature vector $\boldsymbol{x}$, is spam or not. The training data consists of the feature vectors of a number of different email examples as well as the corresponding labels (spam or not spam). For instance, a feature vector could consist of the number of times sales-pitch words like "free", "sale", or "miss out" occur within a given email.

As seen from the above discussion, most questions of interest in supervised learning can be answered if we know the conditional pdf $f(y \mid \boldsymbol{x})$, because we can then in principle work out the function value $g^*(\boldsymbol{x})$.

UNSUPERVISED LEARNING

In contrast, *unsupervised learning* makes no distinction between response and explanatory variables, and the objective is simply to learn the structure of the unknown distribution of the data. In other words, we need to learn $f(\boldsymbol{x})$. In this case the guess $g(\boldsymbol{x})$ is an approximation of $f(\boldsymbol{x})$ and the risk is of the form

$$\ell(g) = \mathbb{E}\,\mathrm{Loss}(f(\boldsymbol{X}), g(\boldsymbol{X})).$$

An example of unsupervised learning is when we wish to analyze the purchasing behaviors of the customers of a grocery shop that has a total of, say, a hundred items on sale. A feature vector here could be a binary vector $\boldsymbol{x} \in \{0,1\}^{100}$ representing the items bought by a customer on a visit to the shop (a 1 in the k-th position if a customer bought item $k \in \{1, \ldots, 100\}$ and a 0 otherwise). Based on a training set $\tau = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, we wish to find any interesting or unusual purchasing patterns. In general, it is difficult to know if an unsupervised learner is doing a good job, because there is no teacher to provide examples of accurate predictions.

The main methodologies for unsupervised learning include *clustering*, *principal component analysis*, and *kernel density estimation*, which will be discussed in Chapter 4.

☞ 121

In the next three sections we will focus on supervised learning. The main supervised learning methodologies are *regression* and *classification*, to be discussed in detail in Chapters 5 and 7. More advanced supervised learning techniques, including *reproducing kernel Hilbert spaces*, *tree methods*, and *deep learning*, will be discussed in Chapters 6, 8, and 9.

☞ 167
☞ 253

2.3 Training and Test Loss

Given an arbitrary prediction function g, it is typically not possible to compute its risk $\ell(g)$ in (2.1). However, using the training sample $\mathcal{T}$, we can approximate $\ell(g)$ via the empirical (sample average) risk

$$\ell_{\mathcal{T}}(g) = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}(Y_i, g(\boldsymbol{X}_i)), \tag{2.3}$$

which we call the *training loss*. The training loss is thus an unbiased estimator of the risk (the expected loss) for a prediction function g, based on the training data. TRAINING LOSS

To approximate the optimal prediction function g^* (the minimizer of the risk $\ell(g)$) we first select a suitable collection of approximating functions $\mathcal{G}$ and then take our *learner* to be the function in $\mathcal{G}$ that minimizes the training loss; that is,

$$g_{\mathcal{T}}^{\mathcal{G}} = \operatorname*{argmin}_{g\in\mathcal{G}} \ell_{\mathcal{T}}(g). \tag{2.4}$$

For example, the simplest and most useful $\mathcal{G}$ is the set of *linear* functions of $\boldsymbol{x}$; that is, the set of all functions $g : \boldsymbol{x} \mapsto \boldsymbol{\beta}^\top \boldsymbol{x}$ for some real-valued vector $\boldsymbol{\beta}$.

We suppress the superscript $\mathcal{G}$ when it is clear which function class is used. Note that minimizing the training loss over all possible functions g (rather than over all $g \in \mathcal{G}$) does not lead to a meaningful optimization problem, as any function g for which $g(\boldsymbol{X}_i) = Y_i$ for all i gives minimal training loss. In particular, for a squared-error loss, the training loss will be 0. Unfortunately, such functions have a poor ability to predict new (that is, independent from $\mathcal{T}$) pairs of data. This poor generalization performance is called *overfitting*. OVERFITTING

By choosing g a function that predicts the training data exactly (and is, for example, 0 otherwise), the squared-error training loss is zero. Minimizing the training loss is not the ultimate goal!

The prediction accuracy of new pairs of data is measured by the *generalization risk* of the learner. For a *fixed* training set τ it is defined as GENERALIZATION RISK

$$\ell(g_\tau^{\mathcal{G}}) = \mathbb{E}\,\text{Loss}(Y, g_\tau^{\mathcal{G}}(\boldsymbol{X})), \tag{2.5}$$

where $(\boldsymbol{X}, Y)$ is distributed according to $f(\boldsymbol{x}, y)$. In the discrete case the generalization risk is therefore: $\ell(g_\tau^{\mathcal{G}}) = \sum_{\boldsymbol{x},y} \text{Loss}(y, g_\tau^{\mathcal{G}}(\boldsymbol{x}))f(\boldsymbol{x}, y)$ (replace the sum with an integral for the continuous case). The situation is illustrated in Figure 2.1, where the distribution of $(\boldsymbol{X}, Y)$ is indicated by the red dots. The training set (points in the shaded regions) determines a fixed prediction function shown as a straight line. Three possible outcomes of $(\boldsymbol{X}, Y)$ are shown (black dots). The amount of loss for each point is shown as the length of the dashed lines. The generalization risk is the average loss over all possible pairs $(\boldsymbol{x}, y)$, weighted by the corresponding $f(\boldsymbol{x}, y)$.

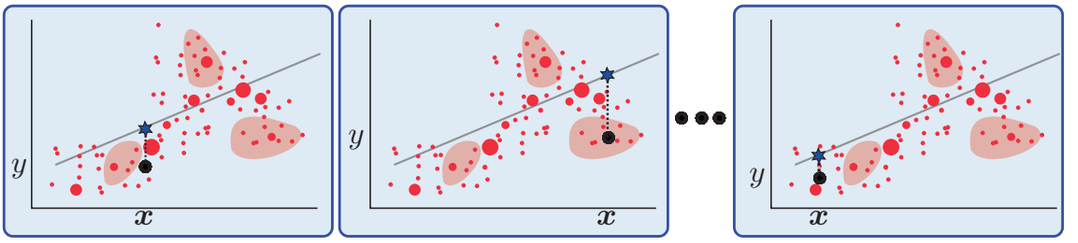

Figure 2.1: The generalization risk for a fixed training set is the weighted-average loss over all possible pairs $(\boldsymbol{x}, y)$.

For a *random* training set $\mathcal{T}$, the generalization risk is thus a random variable that depends on $\mathcal{T}$ (and $\mathcal{G}$). If we average the generalization risk over all possible instances of $\mathcal{T}$, we obtain the *expected generalization risk*:

EXPECTED GENERALIZATION RISK

$$\mathbb{E}\,\ell(g_{\mathcal{T}}^{\mathcal{G}}) = \mathbb{E}\,\mathrm{Loss}(Y, g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{X})), \tag{2.6}$$

where $(\boldsymbol{X}, Y)$ in the expectation above is independent of $\mathcal{T}$. In the discrete case, we have $\mathbb{E}\ell(g_{\mathcal{T}}^{\mathcal{G}}) = \sum_{\boldsymbol{x},y,\boldsymbol{x}_1,y_1,\ldots,\boldsymbol{x}_n,y_n} \mathrm{Loss}(y, g_{\tau}^{\mathcal{G}}(\boldsymbol{x}))f(\boldsymbol{x}, y)f(\boldsymbol{x}_1, y_1)\cdots f(\boldsymbol{x}_n, y_n)$. Figure 2.2 gives an illustration.

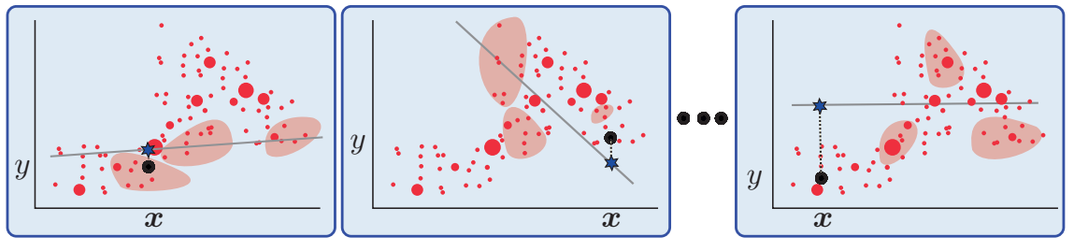

Figure 2.2: The expected generalization risk is the weighted-average loss over all possible pairs $(\boldsymbol{x}, y)$ and over all training sets.

For any outcome τ of the training data, we can estimate the generalization risk without bias by taking the sample average

$$\ell_{\mathcal{T}'}(g_{\tau}^{\mathcal{G}}) := \frac{1}{n'}\sum_{i=1}^{n'} \mathrm{Loss}(Y_i', g_{\tau}^{\mathcal{G}}(\boldsymbol{X}_i')), \tag{2.7}$$

TEST SAMPLE

TEST LOSS

where $\{(\boldsymbol{X}_1', Y_1'), \ldots, (\boldsymbol{X}_{n'}', Y_{n'}')\} =: \mathcal{T}'$ is a so-called *test sample*. The test sample is completely separate from $\mathcal{T}$, but is drawn in the same way as $\mathcal{T}$; that is, via independent draws from $f(\boldsymbol{x}, y)$, for some sample size n'. We call the estimator (2.7) the *test loss*. For a random training set $\mathcal{T}$ we can define $\ell_{\mathcal{T}'}(g_{\mathcal{T}}^{\mathcal{G}})$ similarly. It is then crucial to assume that $\mathcal{T}$ is independent of $\mathcal{T}'$. Table 2.1 summarizes the main definitions and notation for supervised learning.

Table 2.1: Summary of definitions for supervised learning.

$\boldsymbol{x}$	Fixed explanatory (feature) vector.
$\boldsymbol{X}$	Random explanatory (feature) vector.
y	Fixed (real-valued) response.
Y	Random response.
$f(\boldsymbol{x}, y)$	Joint pdf of $\boldsymbol{X}$ and Y, evaluated at $(\boldsymbol{x}, y)$.
$f(y \mid \boldsymbol{x})$	Conditional pdf of Y given $\boldsymbol{X} = \boldsymbol{x}$, evaluated at y.
τ or τ_n	Fixed training data $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$.
$\mathcal{T}$ or $\mathcal{T}_n$	Random training data $\{(\boldsymbol{X}_i, Y_i), i = 1, \ldots, n\}$.
$\mathbf{X}$	Matrix of explanatory variables, with n rows $\boldsymbol{x}_i^\top, i = 1, \ldots, n$ and $\dim(\boldsymbol{x})$ feature columns; one of the features may be the constant 1.
$\boldsymbol{y}$	Vector of response variables $(y_1, \ldots, y_n)^\top$.
g	Prediction (guess) function.
$\text{Loss}(y, \widehat{y})$	Loss incurred when predicting response y with $\widehat{y}$.
$\ell(g)$	Risk for prediction function g; that is, $\mathbb{E}\,\text{Loss}(Y, g(\boldsymbol{X}))$.
g^*	Optimal prediction function; that is, $\operatorname{argmin}_g \ell(g)$.
$g^{\mathcal{G}}$	Optimal prediction function in function class $\mathcal{G}$; that is, $\operatorname{argmin}_{g \in \mathcal{G}} \ell(g)$.
$\ell_\tau(g)$	Training loss for prediction function g; that is, the sample average estimate of $\ell(g)$ based on a fixed training sample τ.
$\ell_{\mathcal{T}}(g)$	The same as $\ell_\tau(g)$, but now for a random training sample $\mathcal{T}$.
$g_\tau^{\mathcal{G}}$ or g_τ	The *learner*: $\operatorname{argmin}_{g \in \mathcal{G}} \ell_\tau(g)$. That is, the optimal prediction function based on a fixed training set τ and function class $\mathcal{G}$. We suppress the superscript $\mathcal{G}$ if the function class is implicit.
$g_{\mathcal{T}}^{\mathcal{G}}$ or $g_{\mathcal{T}}$	The learner, where we have replaced τ with a random training set $\mathcal{T}$.

To compare the predictive performance of various learners in the function class $\mathcal{G}$, as measured by the test loss, we can use the *same* fixed training set τ and test set τ' for all learners. When there is an abundance of data, the "overall" data set is usually (randomly) divided into a training and test set, as depicted in Figure 2.3. We then use the training data to construct various learners $g_\tau^{\mathcal{G}_1}, g_\tau^{\mathcal{G}_2}, \ldots$, and use the test data to select the best (with the smallest test loss) among these learners. In this context the test set is called the *validation set*. Once the best learner has been chosen, a third "test" set can be used to assess the predictive performance of the best learner. The training, validation, and test sets can again be obtained from the overall data set via a random allocation. When the overall data set is of modest size, it is customary to perform the validation phase (model selection) on the training set only, using cross-validation. This is the topic of Section 2.5.2.

VALIDATION SET

☞ 37

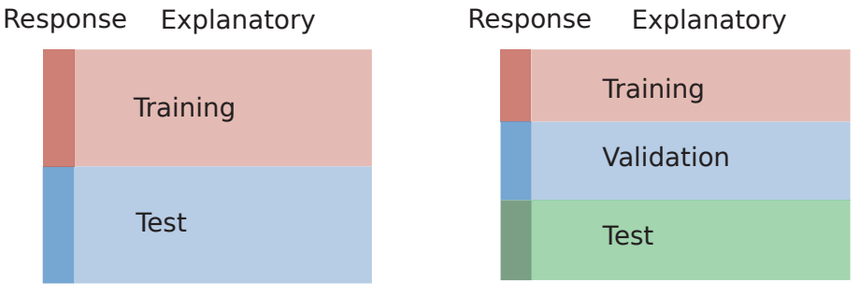

Figure 2.3: Statistical learning algorithms often require the data to be divided into training and test data. If the latter is used for model selection, a third set is needed for testing the performance of the selected model.

We next consider a concrete example that illustrates the concepts introduced so far.

■ **Example 2.1 (Polynomial Regression)** In what follows, it will appear that we have arbitrarily replaced the symbols $\boldsymbol{x}, g, \mathcal{G}$ with $u, h, \mathcal{H}$, respectively. The reason for this switch of notation will become clear at the end of the example.

The data (depicted as dots) in Figure 2.4 are $n = 100$ points $(u_i, y_i), i = 1, \ldots, n$ drawn from iid random points $(U_i, Y_i), i = 1, \ldots, n$, where the $\{U_i\}$ are uniformly distributed on the interval $(0, 1)$ and, given $U_i = u_i$, the random variable Y_i has a normal distribution with expectation $10 - 140u_i + 400u_i^2 - 250u_i^3$ and variance $\ell^* = 25$. This is an example of a *polynomial regression model*. Using a squared-error loss, the optimal prediction function $h^*(u) = \mathbb{E}[Y \mid U = u]$ is thus

POLYNOMIAL REGRESSION MODEL

$$h^*(u) = 10 - 140u + 400u^2 - 250u^3,$$

which is depicted by the dashed curve in Figure 2.4.

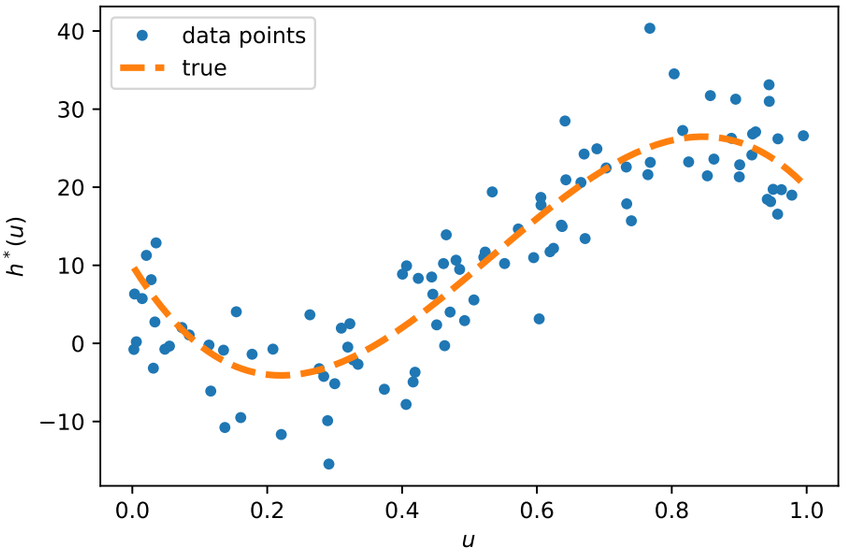

Figure 2.4: Training data and the optimal polynomial prediction function h^*.

To obtain a good estimate of $h^*(u)$ based on the training set $\tau = \{(u_i, y_i), i = 1, \ldots, n\}$, we minimize the outcome of the training loss (2.3):

$$\ell_\tau(h) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(u_i))^2, \tag{2.8}$$

over a suitable set $\mathcal{H}$ of candidate functions. Let us take the set $\mathcal{H}_p$ of polynomial functions in u of order $p-1$:

$$h(u) := \beta_1 + \beta_2 u + \beta_3 u^2 + \cdots + \beta_p u^{p-1} \tag{2.9}$$

for $p = 1, 2, \ldots$ and parameter vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_p]^\top$. This function class contains the best possible $h^*(u) = \mathbb{E}[Y \mid U = u]$ for $p \geqslant 4$. Note that optimization over $\mathcal{H}_p$ is a parametric optimization problem, in that we need to find the best $\boldsymbol{\beta}$. Optimization of (2.8) over $\mathcal{H}_p$ is not straightforward, unless we notice that (2.9) is a *linear* function in $\boldsymbol{\beta}$. In particular, if we map each feature u to a feature vector $\boldsymbol{x} = [1, u, u^2, \ldots, u^{p-1}]^\top$, then the right-hand side of (2.9) can be written as the function

$$g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta},$$

which is linear in $\boldsymbol{x}$ (as well as $\boldsymbol{\beta}$). The optimal $h^*(u)$ in $\mathcal{H}_p$ for $p \geqslant 4$ then corresponds to the function $g^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}^*$ in the set $\mathcal{G}_p$ of linear functions from $\mathbb{R}^p$ to $\mathbb{R}$, where $\boldsymbol{\beta}^* = [10, -140, 400, -250, 0, \ldots, 0]^\top$. Thus, instead of working with the set $\mathcal{H}_p$ of polynomial functions we may prefer to work with the set $\mathcal{G}_p$ of linear functions. This brings us to a very important idea in statistical learning:

Expand the feature space to obtain a *linear* prediction function.

Let us now reformulate the learning problem in terms of the new explanatory (feature) variables $\boldsymbol{x}_i = [1, u_i, u_i^2, \ldots, u_i^{p-1}]^\top$, $i = 1, \ldots, n$. It will be convenient to arrange these feature vectors into a matrix $\mathbf{X}$ with rows $\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top$:

$$\mathbf{X} = \begin{bmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^{p-1} \\ 1 & u_2 & u_2^2 & \cdots & u_2^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_n & u_n^2 & \cdots & u_n^{p-1} \end{bmatrix}. \tag{2.10}$$

Collecting the responses $\{y_i\}$ into a column vector $\boldsymbol{y}$, the training loss (2.3) can now be written compactly as

$$\frac{1}{n}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2. \tag{2.11}$$

To find the optimal learner (2.4) in the class $\mathcal{G}_p$ we need to find the minimizer of (2.11):

$$\widehat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2, \tag{2.12}$$

which is called the *ordinary least-squares* solution. As is illustrated in Figure 2.5, to find $\widehat{\boldsymbol{\beta}}$, we choose $\mathbf{X}\widehat{\boldsymbol{\beta}}$ to be equal to the orthogonal projection of $\boldsymbol{y}$ onto the linear space spanned by the columns of the matrix $\mathbf{X}$; that is, $\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{P}\boldsymbol{y}$, where $\mathbf{P}$ is the *projection matrix*.

ORDINARY LEAST-SQUARES

PROJECTION MATRIX

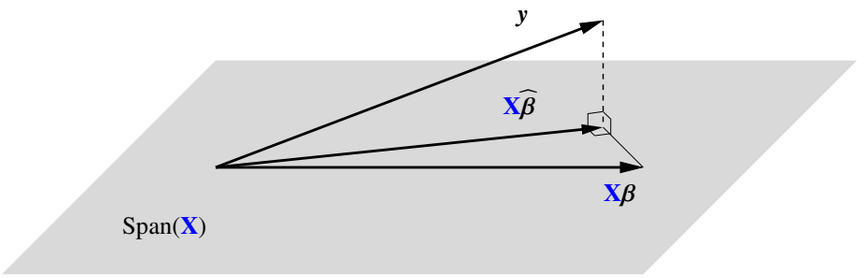

Figure 2.5: $\mathbf{X}\widehat{\boldsymbol{\beta}}$ is the orthogonal projection of $\boldsymbol{y}$ onto the linear space spanned by the columns of the matrix $\mathbf{X}$.

☞ 364

According to Theorem A.4, the projection matrix is given by

$$\mathbf{P} = \mathbf{X}\,\mathbf{X}^{+}, \tag{2.13}$$

☞ 362

PSEUDO-INVERSE

☞ 358

where the $p \times n$ matrix $\mathbf{X}^{+}$ in (2.13) is the *pseudo-inverse* of $\mathbf{X}$. If $\mathbf{X}$ happens to be of *full column rank* (so that none of the columns can be expressed as a linear combination of the other columns), then $\mathbf{X}^{+} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}$.

In any case, from $\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{P}\boldsymbol{y}$ and $\mathbf{P}\mathbf{X} = \mathbf{X}$, we can see that $\widehat{\boldsymbol{\beta}}$ satisfies the *normal equations*:

NORMAL EQUATIONS

$$\mathbf{X}^{\top}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^{\top}\mathbf{P}\boldsymbol{y} = (\mathbf{P}\mathbf{X})^{\top}\boldsymbol{y} = \mathbf{X}^{\top}\boldsymbol{y}. \tag{2.14}$$

This is a set of linear equations, which can be solved very fast and whose solution can be written explicitly as:

$$\widehat{\boldsymbol{\beta}} = \mathbf{X}^{+}\boldsymbol{y}. \tag{2.15}$$

Figure 2.6 shows the trained learners for various values of p:

$$h_{\tau}^{\mathcal{H}_p}(u) = g_{\tau}^{\mathcal{G}_p}(\boldsymbol{x}) = \boldsymbol{x}^{\top}\widehat{\boldsymbol{\beta}}$$

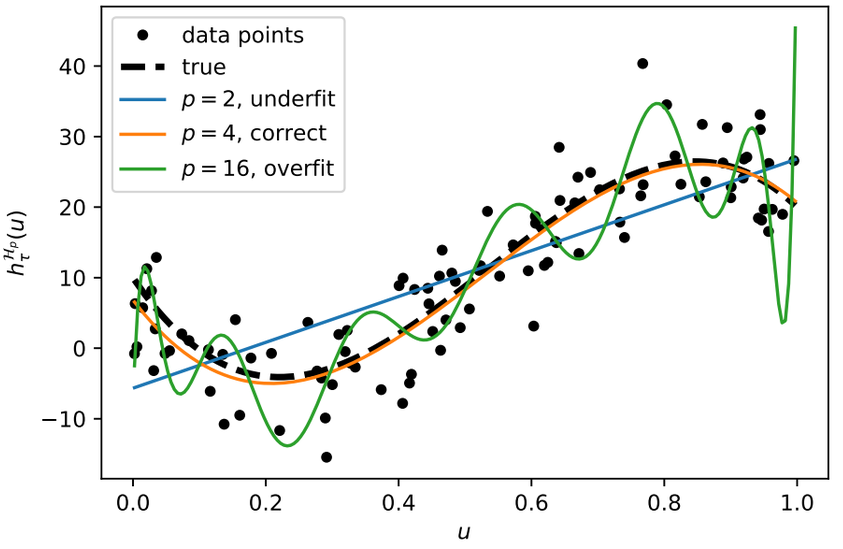

Figure 2.6: Training data with fitted curves for $p = 2, 4$, and 16. The true cubic polynomial curve for $p = 4$ is also plotted (dashed line).

We see that for $p = 16$ the fitted curve lies closer to the data points, but is further away from the dashed true polynomial curve, indicating that we overfit. The choice $p = 4$ (the true cubic polynomial) is much better than $p = 16$, or indeed $p = 2$ (straight line).

Each function class $\mathcal{G}_p$ gives a different learner $g_\tau^{\mathcal{G}_p}$, $p = 1, 2, \ldots$. To assess which is better, we should not simply take the one that gives the smallest training loss. We can always get a *zero* training loss by taking $p = n$, because for any set of n points there exists a polynomial of degree $n - 1$ that interpolates all points!

Instead, we assess the predictive performance of the learners using the test loss (2.7), computed from a test data set. If we collect all n' test feature vectors in a matrix $\mathbf{X}'$ and the corresponding test responses in a vector $\boldsymbol{y}'$, then, similar to (2.11), the test loss can be written compactly as

$$\ell_{\tau'}(g_\tau^{\mathcal{G}_p}) = \frac{1}{n'} \|\boldsymbol{y}' - \mathbf{X}'\widehat{\boldsymbol{\beta}}\|^2,$$

where $\widehat{\boldsymbol{\beta}}$ is given by (2.15), using the training data.

Figure 2.7 shows a plot of the test loss against the number of parameters in the vector $\boldsymbol{\beta}$; that is, p. The graph has a characteristic "bath-tub" shape and is at its lowest for $p = 4$, correctly identifying the polynomial order 3 for the true model. Note that the test loss, as an estimate for the generalization risk (2.7), becomes numerically unreliable after $p = 16$ (the graph goes down, where it should go up). The reader may check that the graph for the training loss exhibits a similar numerical instability for large p, and in fact fails to numerically decrease to 0 for large p, contrary to what it should do in theory. The numerical problems arise from the fact that for large p the columns of the (Vandermonde) matrix $\mathbf{X}$ are of vastly different magnitudes and so floating point errors quickly become very large.

Finally, observe that the lower bound for the test loss is here around 21, which corresponds to an estimate of the minimal (squared-error) risk $\ell^* = 25$.

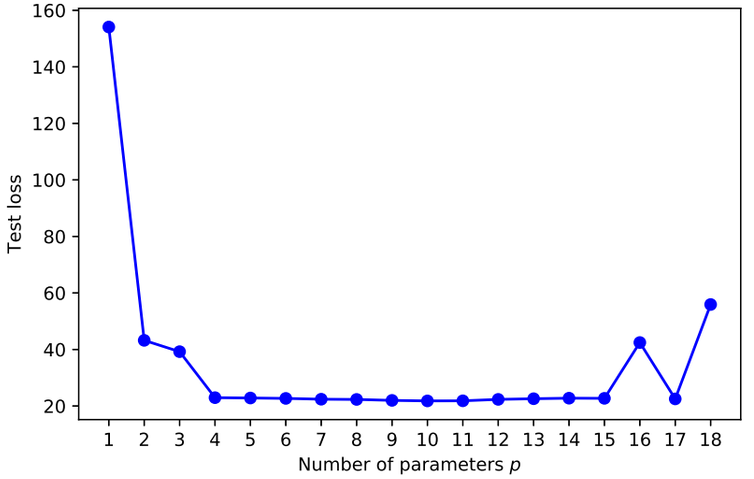

Figure 2.7: Test loss as function of the number of parameters p of the model.

This script shows how the training data were generated and plotted in Python:

polyreg1.py

```
import numpy as np
from numpy.random import rand , randn
from numpy.linalg import norm , solve
import matplotlib.pyplot as plt
def generate_data(beta , sig, n):
   u = np.random.rand(n, 1)
   y = (u ** np.arange(0, 4)) @ beta + sig * np.random.randn(n, 1)
   return u, y

np.random.seed(12)
beta = np.array([[10, -140, 400, -250]]).T
n = 100
sig = 5
u, y = generate_data(beta , sig, n)
xx = np.arange(np.min(u), np.max(u)+5e-3, 5e-3)
yy = np.polyval(np.flip(beta), xx)
plt.plot(u, y, '.', markersize=8)
plt.plot(xx, yy, '--',linewidth=3)
plt.xlabel(r'$u$')
plt.ylabel(r'$h^*(u)$')
plt.legend(['data points','true'])
plt.show()
```

The following code, which imports the code above, fits polynomial models with $p = 1, \ldots, K = 18$ parameters to the training data and plots a selection of fitted curves, as shown in Figure 2.6.

polyreg2.py

```
from polyreg1 import *

max_p = 18
p_range = np.arange(1, max_p + 1, 1)
X = np.ones((n, 1))
betahat, trainloss = {}, {}

for p in p_range:  # p is the number of parameters
    if p > 1:
        X = np.hstack((X, u**(p-1)))  # add column to matrix

    betahat[p] = solve(X.T @ X, X.T @ y)
    trainloss[p] = (norm(y - X @ betahat[p])**2/n)

p = [2, 4, 16]  # select three curves

#replot the points and true line and store in the list "plots"
plots = [plt.plot(u, y, 'k.', markersize=8)[0],
         plt.plot(xx, yy, 'k--',linewidth=3)[0]]
# add the three curves
for i in p:
    yy = np.polyval(np.flip(betahat[i]), xx)
    plots.append(plt.plot(xx, yy)[0])
```

```
plt.xlabel(r'$u$')
plt.ylabel(r'$h^{\mathcal{H}_p}_{\tau}(u)$')
plt.legend(plots,('data points', 'true','$p=2$, underfit',
                  '$p=4$, correct','$p=16$, overfit'))
plt.savefig('polyfitpy.pdf',format='pdf')
plt.show()
```

The last code snippet which imports the previous code, generates the test data and plots the graph of the test loss, as shown in Figure 2.7.

polyreg3.py

```
from polyreg2 import *

# generate test data
u_test, y_test = generate_data(beta, sig, n)

MSE = []
X_test = np.ones((n, 1))

for p in p_range:
    if p > 1:
        X_test = np.hstack((X_test, u_test**(p-1)))

    y_hat = X_test @ betahat[p]  # predictions
    MSE.append(np.sum((y_test - y_hat)**2/n))

plt.plot(p_range, MSE, 'b', p_range, MSE, 'bo')
plt.xticks(ticks=p_range)
plt.xlabel('Number of parameters $p$')
plt.ylabel('Test loss')
```

2.4 Tradeoffs in Statistical Learning

The art of machine learning in the supervised case is to make the generalization risk (2.5) or expected generalization risk (2.6) as small as possible, while using as few computational resources as possible. In pursuing this goal, a suitable class $\mathcal{G}$ of prediction functions has to be chosen. This choice is driven by various factors, such as

- the complexity of the class (e.g., is it rich enough to adequately approximate, or even contain, the optimal prediction function g^*?),
- the ease of training the learner via the optimization program (2.4),
- how accurately the training loss (2.3) estimates the risk (2.1) within class $\mathcal{G}$,
- the feature types (categorical, continuous, etc.).

As a result, the choice of a suitable function class $\mathcal{G}$ usually involves a tradeoff between conflicting factors. For example, a learner from a simple class $\mathcal{G}$ can be trained very

quickly, but may not approximate g^* very well, whereas a learner from a rich class $\mathcal{G}$ that contains g^* may require a lot of computing resources to train.

To better understand the relation between model complexity, computational simplicity, and estimation accuracy, it is useful to decompose the generalization risk into several parts, so that the tradeoffs between these parts can be studied. We will consider two such decompositions: the approximation–estimation tradeoff and the bias–variance tradeoff.

We can decompose the generalization risk (2.5) into the following three components:

$$\ell(g_\tau^{\mathcal{G}}) = \underbrace{\ell^*}_{\text{irreducible risk}} + \underbrace{\ell(g^{\mathcal{G}}) - \ell^*}_{\text{approximation error}} + \underbrace{\ell(g_\tau^{\mathcal{G}}) - \ell(g^{\mathcal{G}})}_{\text{statistical error}}, \tag{2.16}$$

IRREDUCIBLE RISK

where $\ell^* := \ell(g^*)$ is the *irreducible risk* and $g^{\mathcal{G}} := \operatorname{argmin}_{g \in \mathcal{G}} \ell(g)$ is the best learner within class $\mathcal{G}$. No learner can predict a new response with a smaller risk than ℓ^*.

APPROXIMATION ERROR

The second component is the *approximation error*; it measures the difference between the irreducible risk and the best possible risk that can be obtained by selecting the best prediction function in the selected class of functions $\mathcal{G}$. Determining a suitable class $\mathcal{G}$ and minimizing $\ell(g)$ over this class is purely a problem of numerical and functional analysis, as the training data τ are not present. For a fixed $\mathcal{G}$ that does not contain the optimal g^*, the approximation error cannot be made arbitrarily small and may be the dominant component in the generalization risk. The only way to reduce the approximation error is by expanding the class $\mathcal{G}$ to include a larger set of possible functions.

STATISTICAL (ESTIMATION) ERROR

The third component is the *statistical (estimation) error*. It depends on the training set τ and, in particular, on how well the learner $g_\tau^{\mathcal{G}}$ estimates the best possible prediction function, $g^{\mathcal{G}}$, within class $\mathcal{G}$. For any sensible estimator this error should decay to zero (in probability or expectation) as the training size tends to infinity.

☞ 441

APPROXIMATION–ESTIMATION TRADEOFF

The *approximation–estimation tradeoff* pits two competing demands against each other. The first is that the class $\mathcal{G}$ has to be simple enough so that the statistical error is not too large. The second is that the class $\mathcal{G}$ has to be rich enough to ensure a small approximation error. Thus, there is a tradeoff between the approximation and estimation errors.

For the special case of the squared-error loss, the generalization risk is equal to $\ell(g_\tau^{\mathcal{G}}) = \mathbb{E}(Y - g_\tau^{\mathcal{G}}(\boldsymbol{X}))^2$; that is, the expected squared error[1] between the predicted value $g_\tau^{\mathcal{G}}(\boldsymbol{X})$ and the response Y. Recall that in this case the optimal prediction function is given by $g^*(\boldsymbol{x}) = \mathbb{E}[Y \,|\, \boldsymbol{X} = \boldsymbol{x}]$. The decomposition (2.16) can now be interpreted as follows.

1. The first component, $\ell^* = \mathbb{E}(Y - g^*(\boldsymbol{X}))^2$, is the *irreducible error*, as no prediction function will yield a smaller expected squared error.

2. The second component, the approximation error $\ell(g^{\mathcal{G}}) - \ell(g^*)$, is equal to $\mathbb{E}(g^{\mathcal{G}}(\boldsymbol{X}) - g^*(\boldsymbol{X}))^2$. We leave the proof (which is similar to that of Theorem 2.1) as an exercise; see Exercise 2. Thus, the approximation error (defined as a risk difference) can here be interpreted as the expected squared error between the optimal predicted value and the optimal predicted value within the class $\mathcal{G}$.

3. For the third component, the statistical error, $\ell(g_\tau^{\mathcal{G}}) - \ell(g^{\mathcal{G}})$ there is no direct interpretation as an expected squared error *unless* $\mathcal{G}$ is the class of *linear* functions; that is, $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}$ for some vector $\boldsymbol{\beta}$. In this case we can write (see Exercise 3) the statistical error as $\ell(g_\tau^{\mathcal{G}}) - \ell(g^{\mathcal{G}}) = \mathbb{E}(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - g^{\mathcal{G}}(\boldsymbol{X}))^2$.

[1] Colloquially called *mean squared error*.

Thus, when using a squared-error loss, the generalization risk for a linear class $\mathcal{G}$ can be decomposed as:

$$\ell(g_\tau^{\mathcal{G}}) = \mathbb{E}(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - Y)^2 = \ell^* + \underbrace{\mathbb{E}(g^{\mathcal{G}}(\boldsymbol{X}) - g^*(\boldsymbol{X}))^2}_{\text{approximation error}} + \underbrace{\mathbb{E}(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - g^{\mathcal{G}}(\boldsymbol{X}))^2}_{\text{statistical error}}. \tag{2.17}$$

Note that in this decomposition the statistical error is the only term that depends on the training set.

■ **Example 2.2 (Polynomial Regression (cont.))** We continue Example 2.1. Here $\mathcal{G} = \mathcal{G}_p$ is the class of linear functions of $\boldsymbol{x} = [1, u, u^2, \ldots, u^{p-1}]^\top$, and $g^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}^*$. Conditional on $\boldsymbol{X} = \boldsymbol{x}$ we have that $Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$, with $\varepsilon(\boldsymbol{x}) \sim \mathcal{N}(0, \ell^*)$, where $\ell^* = \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 = 25$ is the irreducible error. We wish to understand how the approximation and statistical errors behave as we change the complexity parameter p.

First, we consider the approximation error. Any function $g \in \mathcal{G}_p$ can be written as

$$g(\boldsymbol{x}) = h(u) = \beta_1 + \beta_2 u + \cdots + \beta_p u^{p-1} = [1, u, \ldots, u^{p-1}]\boldsymbol{\beta},$$

and so $g(\boldsymbol{X})$ is distributed as $[1, U, \ldots, U^{p-1}]\boldsymbol{\beta}$, where $U \sim \mathcal{U}(0, 1)$. Similarly, $g^*(\boldsymbol{X})$ is distributed as $[1, U, U^2, U^3]\boldsymbol{\beta}^*$. It follows that an expression for the approximation error is: $\int_0^1 \left([1, u, \ldots, u^{p-1}]\boldsymbol{\beta} - [1, u, u^2, u^3]\boldsymbol{\beta}^*\right)^2 \mathrm{d}u$. To minimize this error, we set the gradient with respect to $\boldsymbol{\beta}$ to zero and obtain the p linear equations ☞ 399

$$\begin{aligned}
\int_0^1 \left([1, u, \ldots, u^{p-1}]\boldsymbol{\beta} - [1, u, u^2, u^3]\boldsymbol{\beta}^*\right) \mathrm{d}u &= 0,\\
\int_0^1 \left([1, u, \ldots, u^{p-1}]\boldsymbol{\beta} - [1, u, u^2, u^3]\boldsymbol{\beta}^*\right) u\,\mathrm{d}u &= 0,\\
&\vdots\\
\int_0^1 \left([1, u, \ldots, u^{p-1}]\boldsymbol{\beta} - [1, u, u^2, u^3]\boldsymbol{\beta}^*\right) u^{p-1}\mathrm{d}u &= 0.
\end{aligned}$$

Let

$$\mathbf{H}_p = \int_0^1 [1, u, \ldots, u^{p-1}]^\top [1, u, \ldots, u^{p-1}]\,\mathrm{d}u$$

be the $p \times p$ *Hilbert matrix*, which has (i, j)-th entry given by $\int_0^1 u^{i+j-2}\,\mathrm{d}u = 1/(i + j - 1)$. Then, the above system of linear equations can be written as $\mathbf{H}_p\boldsymbol{\beta} = \widetilde{\mathbf{H}}\boldsymbol{\beta}^*$, where $\widetilde{\mathbf{H}}$ is the $p \times 4$ upper left sub-block of $\mathbf{H}_{\widetilde{p}}$ and $\widetilde{p} = \max\{p, 4\}$. The solution, which we denote by $\boldsymbol{\beta}_p$, is: HILBERT MATRIX

$$\boldsymbol{\beta}_p = \begin{cases} \frac{65}{6}, & p = 1,\\ [-\frac{20}{3}, 35]^\top, & p = 2,\\ [-\frac{5}{2}, 10, 25]^\top, & p = 3,\\ [10, -140, 400, -250, 0, \ldots, 0]^\top, & p \geqslant 4. \end{cases} \tag{2.18}$$

Hence, the approximation error $\mathbb{E}\left(g^{\mathcal{G}_p}(\boldsymbol{X}) - g^*(\boldsymbol{X})\right)^2$ is given by

$$\int_0^1 \left([1, u, \ldots, u^{p-1}]\boldsymbol{\beta}_p - [1, u, u^2, u^3]\boldsymbol{\beta}^*\right)^2 \mathrm{d}u = \begin{cases} \frac{32225}{252} \approx 127.9, & p = 1,\\ \frac{1625}{63} \approx 25.8, & p = 2,\\ \frac{625}{28} \approx 22.3, & p = 3,\\ 0, & p \geqslant 4. \end{cases} \tag{2.19}$$

Notice how the approximation error becomes smaller as p increases. In this particular example the approximation error is in fact zero for $p \geqslant 4$. In general, as the class of approximating functions $\mathcal{G}$ becomes more complex, the approximation error goes down.

Next, we illustrate the typical behavior of the statistical error. Since $g_\tau(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}$, the statistical error can be written as

$$\int_0^1 \left([1, \ldots, u^{p-1}](\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p)\right)^2 \mathrm{d}u = (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p)^\top \mathbf{H}_p (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p). \tag{2.20}$$

Figure 2.8 illustrates the decomposition (2.17) of the generalization risk for the *same* training set that was used to compute the test loss in Figure 2.7. Recall that test loss gives an estimate of the generalization risk, using independent test data. Comparing the two figures, we see that in this case the two match closely. The global minimum of the statistical error is approximately 0.28, with minimizer $p = 4$. Since the approximation error is monotonically decreasing to zero, $p = 4$ is also the global minimizer of the generalization risk.

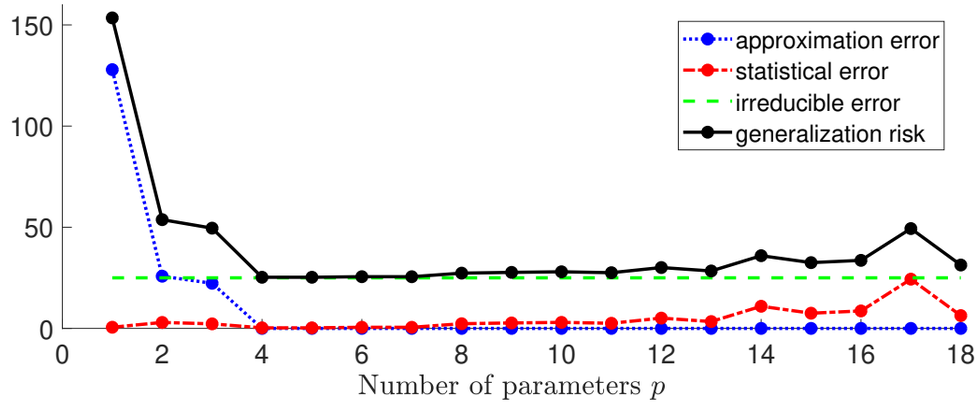

Figure 2.8: The generalization risk for a particular training set is the sum of the irreducible error, the approximation error, and the statistical error. The approximation error decreases to zero as p increases, whereas the statistical error has a tendency to increase after $p = 4$.

Note that the statistical error depends on the estimate $\widehat{\boldsymbol{\beta}}$, which in its turn depends on the training set τ. We can obtain a better understanding of the statistical error by considering its *expected* behavior; that is, averaged over many training sets. This is explored in Exercise 11. ■

Using again a squared-error loss, a second decomposition (for general $\mathcal{G}$) starts from

$$\ell(g_\tau^{\mathcal{G}}) = \ell^* + \ell(g_\tau^{\mathcal{G}}) - \ell(g^*),$$

where the statistical error and approximation error are combined. Using similar reasoning as in the proof of Theorem 2.1, we have

$$\ell(g_\tau^{\mathcal{G}}) = \mathbb{E}(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - Y)^2 = \ell^* + \mathbb{E}\left(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - g^*(\boldsymbol{X})\right)^2 = \ell^* + \mathbb{E}D^2(\boldsymbol{X}, \tau),$$

where $D(\boldsymbol{x}, \tau) := g_\tau^{\mathcal{G}}(\boldsymbol{x}) - g^*(\boldsymbol{x})$. Now consider the random variable $D(\boldsymbol{x}, \mathcal{T})$ for a random training set $\mathcal{T}$. The expectation of its square is:

$$\begin{aligned}\mathbb{E}\left(g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x}) - g^*(\boldsymbol{x})\right)^2 &= \mathbb{E}D^2(\boldsymbol{x}, \mathcal{T}) = (\mathbb{E}D(\boldsymbol{x}, \mathcal{T}))^2 + \mathbb{V}\mathrm{ar}\, D(\boldsymbol{x}, \mathcal{T}) \\ &= \underbrace{(\mathbb{E}g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x}) - g^*(\boldsymbol{x}))^2}_{\text{pointwise squared bias}} + \underbrace{\mathbb{V}\mathrm{ar}\, g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x})}_{\text{pointwise variance}} . \end{aligned} \tag{2.21}$$

If we view the learner $g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x})$ as a function of a random training set, then the *pointwise squared bias* term is a measure for how close $g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x})$ is on average to the true $g^*(\boldsymbol{x})$, whereas the *pointwise variance* term measures the deviation of $g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x})$ from its expected value $\mathbb{E}g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{x})$. The squared bias can be reduced by making the class of functions $\mathcal{G}$ more complex. However, decreasing the bias by increasing the complexity often leads to an increase in the variance term. We are thus seeking learners that provide an optimal balance between the bias and variance, as expressed via a minimal generalization risk. This is called the *bias–variance tradeoff*.

POINTWISE SQUARED BIAS
POINTWISE VARIANCE
BIAS–VARIANCE TRADEOFF

Note that the *expected* generalization risk (2.6) can be written as $\ell^* + \mathbb{E}D^2(\boldsymbol{X}, \mathcal{T})$, where $\boldsymbol{X}$ and $\mathcal{T}$ are independent. It therefore decomposes as

$$\mathbb{E}\,\ell(g_{\mathcal{T}}^{\mathcal{G}}) = \ell^* + \underbrace{\mathbb{E}\,(\mathbb{E}[g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{X}) \,|\, \boldsymbol{X}] - g^*(\boldsymbol{X}))^2}_{\text{expected squared bias}} + \underbrace{\mathbb{E}[\mathbb{V}\mathrm{ar}[g_{\mathcal{T}}^{\mathcal{G}}(\boldsymbol{X}) \,|\, \boldsymbol{X}]]}_{\text{expected variance}} . \tag{2.22}$$

2.5 Estimating Risk

The most straightforward way to quantify the generalization risk (2.5) is to estimate it via the test loss (2.7). However, the generalization risk depends inherently on the training set, and so different training sets may yield significantly different estimates. Moreover, when there is a limited amount of data available, reserving a substantial proportion of the data for testing rather than training may be uneconomical. In this section we consider different methods for estimating risk measures which aim to circumvent these difficulties.

2.5.1 In-Sample Risk

We mentioned that, due to the phenomenon of overfitting, the training loss of the learner, $\ell_\tau(g_\tau)$ (for simplicity, here we omit $\mathcal{G}$ from $g_\tau^{\mathcal{G}}$), is not a good estimate of the generalization risk $\ell(g_\tau)$ of the learner. One reason for this is that we use the same data for both training the model and assessing its risk. How should we then estimate the generalization risk or expected generalization risk?

To simplify the analysis, suppose that we wish to estimate the average accuracy of the predictions of the learner g_τ at the n feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ (these are part of the training set τ). In other words, we wish to estimate the *in-sample risk* of the learner g_τ:

IN-SAMPLE RISK

$$\ell_{\text{in}}(g_\tau) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\,\mathrm{Loss}(Y_i', g_\tau(\boldsymbol{x}_i)), \tag{2.23}$$

where each response Y_i' is drawn from $f(y \,|\, \boldsymbol{x}_i)$, independently. Even in this simplified setting, the training loss of the learner will be a poor estimate of the in-sample risk. Instead, the

proper way to assess the prediction accuracy of the learner at the feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, is to draw new response values $Y_i' \sim f(y \,|\, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, that are independent from the responses $y_1, \ldots, y_n$ in the training data, and then estimate the in-sample risk of g_τ via

$$\frac{1}{n} \sum_{i=1}^{n} \text{Loss}(Y_i', g_\tau(\boldsymbol{x}_i)).$$

For a fixed training set τ, we can compare the training loss of the learner with the in-sample risk. Their difference,

$$\text{op}_\tau = \ell_{\text{in}}(g_\tau) - \ell_\tau(g_\tau),$$

is called the *optimism* (of the training loss), because it measures how much the training loss underestimates (is optimistic about) the unknown in-sample risk. Mathematically, it is simpler to work with the *expected optimism*:

EXPECTED OPTIMISM

$$\mathbb{E}[\text{op}_\mathcal{T} \,|\, \boldsymbol{X}_1 = \boldsymbol{x}_1, \ldots, \boldsymbol{X}_n = \boldsymbol{x}_n] =: \mathbb{E}_\mathbf{X} \, \text{op}_\mathcal{T},$$

where the expectation is taken over a random training set $\mathcal{T}$, conditional on $\boldsymbol{X}_i = \boldsymbol{x}_i$, $i = 1, \ldots, n$. For ease of notation, we have abbreviated the expected optimism to $\mathbb{E}_\mathbf{X} \, \text{op}_\mathcal{T}$, where $\mathbb{E}_\mathbf{X}$ denotes the expectation operator conditional on $\boldsymbol{X}_i = \boldsymbol{x}_i, i = 1, \ldots, n$. As in Example 2.1, the feature vectors are stored as the rows of an $n \times p$ matrix $\mathbf{X}$. It turns out that the expected optimism for various loss functions can be expressed in terms of the (conditional) covariance between the observed and predicted response.

Theorem 2.2: Expected Optimism

For the squared-error loss and 0–1 loss with 0–1 response, the expected optimism is

$$\mathbb{E}_\mathbf{X} \, \text{op}_\mathcal{T} = \frac{2}{n} \sum_{i=1}^{n} \mathbb{C}\text{ov}_\mathbf{X}(g_\mathcal{T}(\boldsymbol{x}_i), Y_i). \tag{2.24}$$

Proof: In what follows, all expectations are taken conditional on $\boldsymbol{X}_1 = \boldsymbol{x}_1, \ldots, \boldsymbol{X}_n = \boldsymbol{x}_n$. Let Y_i be the response for $\boldsymbol{x}_i$ and let $\widehat{Y}_i = g_\mathcal{T}(\boldsymbol{x}_i)$ be the predicted value. Note that the latter depends on $Y_1, \ldots, Y_n$. Also, let Y_i' be an independent copy of Y_i for the *same* $\boldsymbol{x}_i$, as in (2.23). In particular, Y_i' has the same distribution as Y_i and is statistically independent of all $\{Y_j\}$, including Y_i, and therefore is also independent of $\widehat{Y}_i$. We have

$$\begin{aligned}
\mathbb{E}_\mathbf{X} \, \text{op}_\mathcal{T} &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\mathbf{X} \left[(Y_i' - \widehat{Y}_i)^2 - (Y_i - \widehat{Y}_i)^2 \right] = \frac{2}{n} \sum_{i=1}^{n} \mathbb{E}_\mathbf{X} \left[(Y_i - Y_i') \widehat{Y}_i \right] \\
&= \frac{2}{n} \sum_{i=1}^{n} \left(\mathbb{E}_\mathbf{X}[Y_i \widehat{Y}_i] - \mathbb{E}_\mathbf{X} Y_i \, \mathbb{E}_\mathbf{X} \widehat{Y}_i \right) = \frac{2}{n} \sum_{i=1}^{n} \mathbb{C}\text{ov}_\mathbf{X}(\widehat{Y}_i, Y_i).
\end{aligned}$$

The proof for the 0–1 loss with 0–1 response is left as Exercise 4. □

In summary, the expected optimism indicates how much, on average, the training loss deviates from the expected in-sample risk. Since the covariance of independent random variables is zero, the expected optimism is zero if the learner $g_\mathcal{T}$ is statistically independent from the responses $Y_1, \ldots, Y_n$.

■ **Example 2.3 (Polynomial Regression (cont.))** We continue Example 2.2, where the components of the response vector $\boldsymbol{Y} = [Y_1, \ldots, Y_n]^\top$ are independent and normally distributed with variance $\ell^* = 25$ (the irreducible error) and expectations $\mathbb{E}_{\mathbf{X}} Y_i = g^*(\boldsymbol{x}_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta}^*$, $i = 1, \ldots, n$. Using the formula (2.15) for the least-squares estimator $\widehat{\boldsymbol{\beta}}$, the expected optimism (2.24) is

$$\frac{2}{n}\sum_{i=1}^{n} \mathbb{C}\mathrm{ov}_{\mathbf{X}}\left(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}, Y_i\right) = \frac{2}{n}\mathrm{tr}\left(\mathbb{C}\mathrm{ov}_{\mathbf{X}}\left(\mathbf{X}\widehat{\boldsymbol{\beta}}, \boldsymbol{Y}\right)\right) = \frac{2}{n}\mathrm{tr}\left(\mathbb{C}\mathrm{ov}_{\mathbf{X}}\left(\mathbf{X}\mathbf{X}^+\boldsymbol{Y}, \boldsymbol{Y}\right)\right)$$
$$= \frac{2\mathrm{tr}\left(\mathbf{X}\mathbf{X}^+\mathbb{C}\mathrm{ov}_{\mathbf{X}}\left(\boldsymbol{Y}, \boldsymbol{Y}\right)\right)}{n} = \frac{2\ell^*\mathrm{tr}\left(\mathbf{X}\mathbf{X}^+\right)}{n} = \frac{2\ell^* p}{n}.$$

In the last equation we used the cyclic property of the trace (Theorem A.1): $\mathrm{tr}(\mathbf{X}\mathbf{X}^+) = \mathrm{tr}(\mathbf{X}^+\mathbf{X}) = \mathrm{tr}(\mathbf{I}_p)$, assuming that $\mathrm{rank}(\mathbf{X}) = p$. Therefore, an estimate for the in-sample risk (2.23) is: ☞ 359

$$\widehat{\ell}_{\mathrm{in}}(g_\tau) = \ell_\tau(g_\tau) + 2\ell^* p/n, \tag{2.25}$$

where we have assumed that the irreducible risk ℓ^* is known. Figure 2.9 shows that this estimate is very close to the test loss from Figure 2.7. Hence, instead of computing the test loss to assess the best model complexity p, we could simply have minimized the training loss plus the correction term $2\ell^* p/n$. In practice, ℓ^* also has to be estimated somehow.

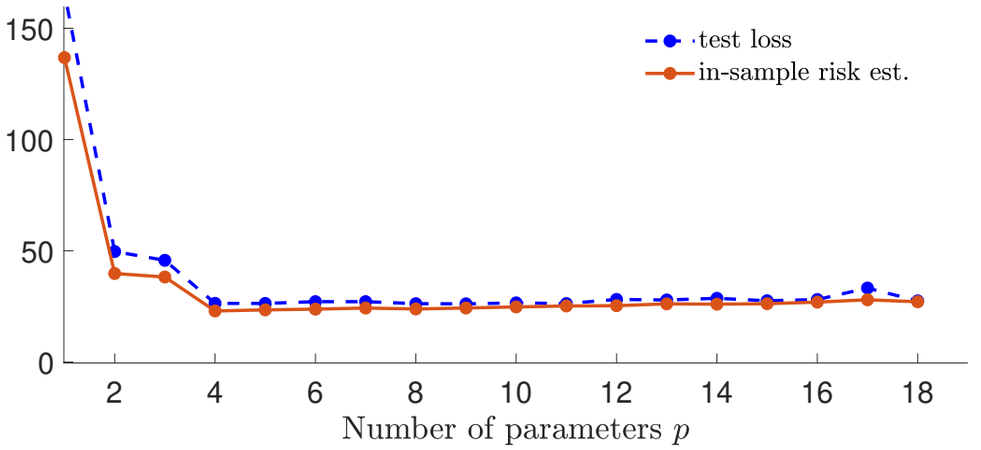

Figure 2.9: In-sample risk estimate $\widehat{\ell}_{\mathrm{in}}(g_\tau)$ as a function of the number of parameters p of the model. The test loss is superimposed as a blue dashed curve.

■

2.5.2 Cross-Validation

In general, for complex function classes $\mathcal{G}$, it is very difficult to derive simple formulas of the approximation and statistical errors, let alone for the generalization risk or expected generalization risk. As we saw, when there is an abundance of data, the easiest way to assess the generalization risk for a given training set τ is to obtain a test set τ' and evaluate the test loss (2.7). When a sufficiently large test set is not available but computational resources are cheap, one can instead gain direct knowledge of the expected generalization risk via a computationally intensive method called *cross-validation*. ☞ 24

CROSS-VALIDATION

The idea is to make multiple identical copies of the data set, and to partition each copy into different training and test sets, as illustrated in Figure 2.10. Here, there are four copies of the data set (consisting of response and explanatory variables). Each copy is divided into a test set (colored blue) and training set (colored pink). For each of these sets, we estimate the model parameters using only training data and then predict the responses for the test set. The average loss between the predicted and observed responses is then a measure for the predictive power of the model.

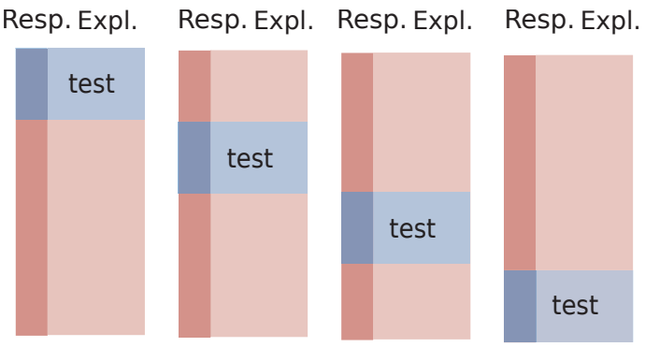

Figure 2.10: An illustration of four-fold cross-validation, representing four copies of the same data set. The data in each copy is partitioned into a training set (pink) and a test set (blue). The darker columns represent the response variable and the lighter ones the explanatory variables.

FOLDS

In particular, suppose we partition a data set $\mathcal{T}$ of size n into K *folds* $C_1, \ldots, C_K$ of sizes $n_1, \ldots, n_K$ (hence, $n_1 + \cdots + n_K = n$). Typically $n_k \approx n/K$, $k = 1, \ldots, K$.

Let ℓ_{C_k} be the test loss when using C_k as test data and all remaining data, denoted $\mathcal{T}_{-k}$, as training data. Each ℓ_{C_k} is an unbiased estimator of the generalization risk for training set $\mathcal{T}_{-k}$; that is, for $\ell(g_{\mathcal{T}_{-k}})$.

K-FOLD CROSS-VALIDATION

The *K-fold cross-validation* loss is the weighted average of these risk estimators:

$$\begin{aligned} \mathrm{CV}_K &= \sum_{k=1}^{K} \frac{n_k}{n} \ell_{C_k}(g_{\mathcal{T}_{-k}}) \\ &= \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k} \mathrm{Loss}(g_{\mathcal{T}_{-k}}(\boldsymbol{x}_i), y_i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \mathrm{Loss}(g_{\mathcal{T}_{-\kappa(i)}}(\boldsymbol{x}_i), y_i), \end{aligned}$$

where the function $\kappa : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ indicates to which of the K folds each of the n observations belongs. As the average is taken over varying training sets $\{\mathcal{T}_{-k}\}$, it estimates the expected generalization risk $\mathbb{E}\,\ell(g_{\mathcal{T}})$, rather than the generalization risk $\ell(g_\tau)$ for the particular training set τ.

■ **Example 2.4 (Polynomial Regression (cont.))** For the polynomial regression example, we can calculate a K-fold cross-validation loss with a nonrandom partitioning of the training set using the following code, which imports the previous code for the polynomial regression example. We omit the full plotting code.

polyregCV.py

```
from polyreg3 import *

K_vals = [5, 10, 100]  # number of folds
cv = np.zeros((len(K_vals), max_p))   # cv loss
X = np.ones((n, 1))

for p in p_range:
  if p > 1:
    X = np.hstack((X, u**(p-1)))
  j = 0
  for K in K_vals:
    loss = []
    for k in range(1, K+1):
       # integer indices of test samples
       test_ind = ((n/K)*(k-1) + np.arange(1,n/K+1)-1).astype('int')
       train_ind = np.setdiff1d(np.arange(n), test_ind)

       X_train, y_train = X[train_ind, :], y[train_ind, :]
       X_test, y_test = X[test_ind, :], y[test_ind]

       # fit model and evaluate test loss
       betahat = solve(X_train.T @ X_train, X_train.T @ y_train)
       loss.append(norm(y_test - X_test @ betahat) ** 2)

    cv[j, p-1] = sum(loss)/n
    j += 1

# basic plotting
plt.plot(p_range, cv[0, :], 'k-.')
plt.plot(p_range, cv[1, :], 'r')
plt.plot(p_range, cv[2, :], 'b--')
plt.show()
```

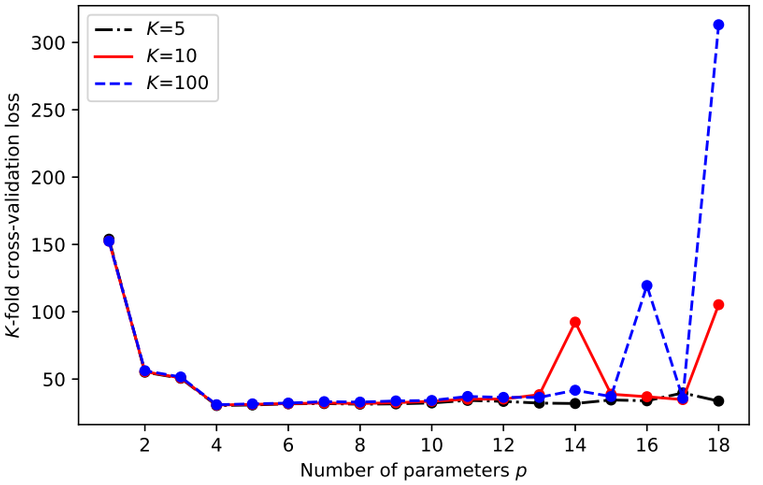

Figure 2.11: K-fold cross-validation for the polynomial regression example.

Figure 2.11 shows the cross-validation loss for $K \in \{5, 10, 100\}$. The case $K = 100$ corresponds to the *leave-one-out cross-validation*, which can be computed more efficiently using the formula in Theorem 5.1. ■

LEAVE-ONE-OUT CROSS-VALIDATION

☞ 174

2.6 Modeling Data

MODEL

The first step in any data analysis is to *model* the data in one form or another. For example, in an *unsupervised* learning setting with data represented by a vector $\boldsymbol{x} = [x_1, \ldots, x_p]^\top$, a very general model is to assume that $\boldsymbol{x}$ is the outcome of a random vector $\boldsymbol{X} = [X_1, \ldots, X_p]^\top$ with some unknown pdf f. The model can then be refined by assuming a specific form of f.

When given a sequence of such data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, one of the simplest models is to assume that the corresponding random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are *independent and identically distributed (iid)*. We write

☞ 431

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{\text{iid}}{\sim} f \quad \text{or} \quad \boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{\text{iid}}{\sim} \mathsf{Dist},$$

to indicate that the random vectors form an iid sample from a sampling pdf f or sampling distribution Dist. This model formalizes the notion that the knowledge about one variable does not provide extra information about another variable. The main theoretical use of independent data models is that the joint density of the random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ is simply the *product* of the marginal ones; see Theorem C.1. Specifically,

☞ 431

$$f_{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = f(\boldsymbol{x}_1) \cdots f(\boldsymbol{x}_n).$$

In most models of this kind, our approximation or model for the sampling distribution is specified up to a small number of parameters. That is, $g(\boldsymbol{x})$ is of the form $g(\boldsymbol{x} \,|\, \boldsymbol{\beta})$ which is known up to some parameter vector $\boldsymbol{\beta}$. Examples for the one-dimensional case ($p = 1$) include the $\mathcal{N}(\mu, \sigma^2)$, $\mathsf{Bin}(n, p)$, and $\mathsf{Exp}(\lambda)$ distributions. See Tables C.1 and C.2 for other common sampling distributions.

☞ 427

Typically, the parameters are unknown and must be estimated from the data. In a nonparametric setting the whole sampling distribution would be unknown. To visualize the underlying sampling distribution from outcomes $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ one can use graphical representations such as histograms, density plots, and empirical cumulative distribution functions, as discussed in Chapter 1.

☞ 11

If the order in which the data were collected (or their labeling) is not informative or relevant, then the joint pdf of $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ satisfies the symmetry:

$$f_{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = f_{\boldsymbol{X}_{\pi_1}, \ldots, \boldsymbol{X}_{\pi_n}}(\boldsymbol{x}_{\pi_1}, \ldots, \boldsymbol{x}_{\pi_n}) \tag{2.26}$$

for any permutation $\pi_1, \ldots, \pi_n$ of the integers $1, \ldots, n$. We say that the infinite sequence $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ is *exchangeable* if this permutational invariance (2.26) holds for any finite subset of the sequence. As we shall see in Section 2.9 on Bayesian learning, it is common to assume that the random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are a subset of an exchangeable sequence and thus satisfy (2.26). Note that while iid random variables are exchangeable, the converse is not necessarily true. Thus, the assumption of an exchangeable sequence of random vectors is weaker than the assumption of iid random vectors.

EXCHANGEABLE

Figure 2.12 illustrates the modeling tradeoffs. The keywords within the triangle represent various modeling paradigms. A few keywords have been highlighted, symbolizing their importance in modeling. The specific meaning of the keywords does not concern us here, but the point is there are many models to choose from, depending on what assumptions are made about the data.

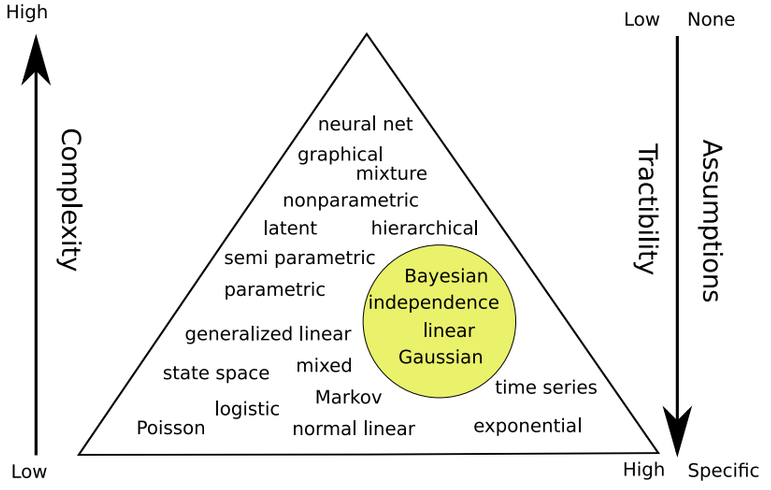

Figure 2.12: Illustration of the modeling dilemma. Complex models are more generally applicable, but may be difficult to analyze. Simple models may be highly tractable, but may not describe the data accurately. The triangular shape signifies that there are a great many specific models but not so many generic ones.

On the one hand, models that make few assumptions are more widely applicable, but at the same time may not be very mathematically tractable or provide insight into the nature of the data. On the other hand, very specific models may be easy to handle and interpret, but may not match the data very well. This tradeoff between the tractability and applicability of the model is very similar to the approximation–estimation tradeoff described in Section 2.4.

In the typical *unsupervised* setting we have a training set $\tau = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ that is viewed as the outcome of n iid random variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ from some unknown pdf f. The objective is then to learn or estimate f from the finite training data. To put the learning in a similar framework as for supervised learning discussed in the preceding Sections 2.3–2.5, we begin by specifying a class of probability density functions $\mathcal{G}_p := \{g(\cdot \,|\, \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\boldsymbol{\theta}$ is a parameter in some subset $\boldsymbol{\Theta}$ of $\mathbb{R}^p$. We now seek the best g in $\mathcal{G}_p$ to minimize some risk. Note that $\mathcal{G}_p$ may not necessarily contain the true f even for very large p.

We stress that our notation $g(\boldsymbol{x})$ has a different meaning in the supervised and unsupervised case. In the supervised case, g is interpreted as a prediction function for a response y; in the unsupervised setting, g is an approximation of a density f.

For each $\boldsymbol{x}$ we measure the discrepancy between the true model $f(\boldsymbol{x})$ and the hypothesized model $g(\boldsymbol{x} \,|\, \boldsymbol{\theta})$ using the loss function

$$\text{Loss}(f(\boldsymbol{x}), g(\boldsymbol{x} \,|\, \boldsymbol{\theta})) = \ln \frac{f(\boldsymbol{x})}{g(\boldsymbol{x} \,|\, \boldsymbol{\theta})} = \ln f(\boldsymbol{x}) - \ln g(\boldsymbol{x} \,|\, \boldsymbol{\theta}).$$

The expected value of this loss (that is, the risk) is thus

$$\ell(g) = \mathbb{E} \ln \frac{f(\boldsymbol{X})}{g(\boldsymbol{X} \,|\, \boldsymbol{\theta})} = \int f(\boldsymbol{x}) \ln \frac{f(\boldsymbol{x})}{g(\boldsymbol{x} \,|\, \boldsymbol{\theta})} \, \mathrm{d}\boldsymbol{x}. \tag{2.27}$$

KULLBACK–LEIBLER DIVERGENCE

The integral in (2.27) provides a fundamental way to measure the distance between two densities and is called the *Kullback–Leibler (KL) divergence*[2] between f and $g(\cdot \,|\, \boldsymbol{\theta})$. Note that the KL divergence is not symmetric in f and $g(\cdot \,|\, \boldsymbol{\theta})$. Moreover, it is always greater than or equal to 0 (see Exercise 15) and equal to 0 when $f = g(\cdot \,|\, \boldsymbol{\theta})$.

Using similar notation as for the supervised learning setting in Table 2.1, define $g^{\mathcal{G}_p}$ as the global minimizer of the risk in the class $\mathcal{G}_p$; that is, $g^{\mathcal{G}_p} = \operatorname{argmin}_{g \in \mathcal{G}_p} \ell(g)$. If we define

$$\begin{aligned} \boldsymbol{\theta}^* &= \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E} \operatorname{Loss}(f(\boldsymbol{X}), g(\boldsymbol{X} \,|\, \boldsymbol{\theta})) = \operatorname*{argmin}_{\boldsymbol{\theta}} \int (\ln f(\boldsymbol{x}) - \ln g(\boldsymbol{x} \,|\, \boldsymbol{\theta})) f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \int f(\boldsymbol{x}) \ln g(\boldsymbol{x} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathbb{E} \ln g(\boldsymbol{X} \,|\, \boldsymbol{\theta}), \end{aligned}$$

then $g^{\mathcal{G}_p} = g(\cdot \,|\, \boldsymbol{\theta}^*)$ and learning $g^{\mathcal{G}_p}$ is equivalent to learning (or estimating) $\boldsymbol{\theta}^*$. To learn $\boldsymbol{\theta}^*$ from a training set $\tau = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ we then minimize the training loss,

$$\frac{1}{n} \sum_{i=1}^{n} \operatorname{Loss}(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) = -\frac{1}{n} \sum_{i=1}^{n} \ln g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}) + \frac{1}{n} \sum_{i=1}^{n} \ln f(\boldsymbol{x}_i),$$

giving:

$$\widehat{\boldsymbol{\theta}}_n := \operatorname*{argmax}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \ln g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}). \tag{2.28}$$

As the logarithm is an increasing function, this is equivalent to

$$\widehat{\boldsymbol{\theta}}_n := \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{n} g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}),$$

where $\prod_{i=1}^{n} g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})$ is the *likelihood* of the data; that is, the joint density of the $\{\boldsymbol{X}_i\}$ evaluated at the points $\{\boldsymbol{x}_i\}$. We therefore have recovered the classical *maximum likelihood estimate* of $\boldsymbol{\theta}^*$.

MAXIMUM LIKELIHOOD ESTIMATE

☞ 458

When the risk $\ell(g(\cdot \,|\, \boldsymbol{\theta}))$ is convex in $\boldsymbol{\theta}$ over a convex set $\boldsymbol{\Theta}$, we can find the maximum likelihood estimator by setting the gradient of the training loss to zero; that is, we solve

$$-\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{S}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}) = \mathbf{0},$$

where $\boldsymbol{S}(\boldsymbol{x} \,|\, \boldsymbol{\theta}) := \frac{\partial \ln g(\boldsymbol{x} \,|\, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the gradient of $\ln g(\boldsymbol{x} \,|\, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ and is often called the *score*.

SCORE

■ **Example 2.5 (Exponential Model)** Suppose we have the training data $\tau_n = \{x_1, \ldots, x_n\}$, which is modeled as a realization of n positive iid random variables: $X_1, \ldots, X_n \sim_{\text{iid}} f(x)$. We select the class of approximating functions $\mathcal{G}$ to be the parametric class $\{g : g(x \,|\, \theta) =$

[2] Sometimes called cross-entropy distance.

$\theta \exp(-x\theta), x > 0, \theta > 0\}$. In other words, we look for the best $g^{\mathcal{G}}$ within the family of exponential distributions with unknown parameter $\theta > 0$. The likelihood of the data is

$$\prod_{i=1}^{n} g(x_i \,|\, \theta) = \prod_{i=1}^{n} \theta \exp(-\theta x_i) = \exp(-\theta\, n\, \overline{x}_n + n \ln \theta)$$

and the score is $S(x \,|\, \theta) = -x + \theta^{-1}$. Thus, maximizing the likelihood with respect to θ is the same as maximizing $-\theta\, n\, \overline{x}_n + n \ln \theta$ or solving $-\sum_{i=1}^{n} S(x_i \,|\, \theta)/n = \overline{x}_n - \theta^{-1} = 0$. In other words, the solution to (2.28) is the maximum likelihood estimate $\widehat{\theta}_n = 1/\overline{x}_n$. ■

In a *supervised* setting, where the data is represented by a vector $\boldsymbol{x}$ of explanatory variables and a response y, the general model is that $(\boldsymbol{x}, y)$ is an outcome of $(\boldsymbol{X}, Y) \sim f$ for some unknown f. And for a training sequence $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ the default model assumption is that $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n) \sim_{\text{iid}} f$. As explained in Section 2.2, the analysis primarily involves the conditional pdf $f(y \,|\, \boldsymbol{x})$ and in particular (when using the squared-error loss) the conditional expectation $g^*(\boldsymbol{x}) = \mathbb{E}[Y \,|\, \boldsymbol{X} = \boldsymbol{x}]$. The resulting representation (2.2) allows us to then write the response at $\boldsymbol{X} = \boldsymbol{x}$ as a function of the feature $\boldsymbol{x}$ plus an error term: $Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$.

This leads to the simplest and most important model for supervised learning, where we choose a *linear* class $\mathcal{G}$ of prediction or guess functions and assume that it is rich enough to contain the true g^*. If we further assume that, conditional on $\boldsymbol{X} = \boldsymbol{x}$, the error term ε does not depend on $\boldsymbol{x}$, that is, $\mathbb{E}\,\varepsilon = 0$ and $\mathbb{V}\text{ar}\,\varepsilon = \sigma^2$, then we obtain the following model.

Definition 2.1: Linear Model

In a *linear model* the response Y depends on a p-dimensional explanatory variable $\boldsymbol{x} = [x_1, \ldots, x_p]^\top$ via the linear relationship

LINEAR MODEL

$$Y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon, \tag{2.29}$$

where $\mathbb{E}\,\varepsilon = 0$ and $\mathbb{V}\text{ar}\,\varepsilon = \sigma^2$.

Note that (2.29) is a model for a single pair $(\boldsymbol{x}, Y)$. The model for the training set $\{(\boldsymbol{x}_i, Y_i)\}$ is simply that each Y_i satisfies (2.29) (with $\boldsymbol{x} = \boldsymbol{x}_i$) and that the $\{Y_i\}$ are independent. Gathering all responses in the vector $\boldsymbol{Y} = [Y_1, \ldots, Y_n]^\top$, we can write

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.30}$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_n]^\top$ is a vector of iid copies of ε and $\mathbf{X}$ is the so-called *model matrix*, with rows $\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top$. Linear models are fundamental building blocks of statistical learning algorithms. For this reason, a large part of Chapter 5 is devoted to linear regression models.

MODEL MATRIX

☞ 167

■ **Example 2.6 (Polynomial Regression (cont.))** For our running Example 2.1, we see that the data is described by a linear model of the form (2.30), with model matrix $\mathbf{X}$ given in (2.10). ■

☞ 26

Before we discuss a few other models in the following sections, we would like to emphasize a number of points about modeling.

- *Any* model for data is likely to be *wrong*. For example, real data (as opposed to computer-generated data) are often assumed to come from a normal distribution, which is never exactly true. However, an important advantage of using a normal distribution is that it has many nice mathematical properties, as we will see in Section 2.7.
- Most data models depend on a number of unknown parameters, which need to be estimated from the observed data.
- Any model for real-life data needs to be *checked* for suitability. An important criterion is that data simulated from the model should resemble the observed data, at least for a certain choice of model parameters.

Here are some guidelines for choosing a model. Think of the data as a spreadsheet or data frame, as in Chapter 1, where rows represent the data units and the columns the data features (variables, groups).

- First establish the *type* of the features (quantitative, qualitative, discrete, continuous, etc.).
- Assess whether the data can be assumed to be independent across rows or columns.
- Decide on the level of generality of the model. For example, should we use a simple model with a few unknown parameters or a more generic model that has a large number of parameters? Simple specific models are easier to fit to the data (low estimation error) than more general models, but the fit itself may not be accurate (high approximation error). The tradeoffs discussed in Section 2.4 play an important role here.
- Decide on using a classical (frequentist) or Bayesian model. Section 2.9 gives a short introduction to Bayesian learning.

☞ 47

2.7 Multivariate Normal Models

A standard model for numerical observations $x_1, \ldots, x_n$ (forming, e.g., a column in a spreadsheet or data frame) is that they are the outcomes of iid normal random variables

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2).$$

It is helpful to view a normally distributed random variable as a simple transformation of a standard normal random variable. To wit, if Z has a standard normal distribution, then $X = \mu + \sigma Z$ has a $\mathcal{N}(\mu, \sigma^2)$ distribution. The generalization to n dimensions is discussed in Appendix C.7. We summarize the main points: Let $Z_1, \ldots, Z_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. The pdf of $\boldsymbol{Z} = [Z_1, \ldots, Z_n]^\top$ (that is, the joint pdf of $Z_1, \ldots, Z_n$) is given by

☞ 436

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2} z_i^2} = (2\pi)^{-\frac{n}{2}} \mathrm{e}^{-\frac{1}{2} \boldsymbol{z}^\top \boldsymbol{z}}, \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{2.31}$$

We write $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and say that $\boldsymbol{Z}$ has a standard normal distribution in $\mathbb{R}^n$. Let

$$\boldsymbol{X} = \boldsymbol{\mu} + \mathbf{B}\,\boldsymbol{Z} \tag{2.32}$$

for some $m \times n$ matrix $\mathbf{B}$ and m-dimensional vector $\boldsymbol{\mu}$. Then $\boldsymbol{X}$ has expectation vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$; see (C.20) and (C.21). This leads to the following definition. ☞ 434

Definition 2.2: Multivariate Normal Distribution

An m-dimensional random vector $\boldsymbol{X}$ that can be written in the form (2.32) for some m-dimensional vector $\boldsymbol{\mu}$ and $m \times n$ matrix $\mathbf{B}$, with $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, is said to have a *multivariate normal* or *multivariate Gaussian* distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$. We write $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

MULTIVARIATE NORMAL

The m-dimensional density of a multivariate normal distribution has a very similar form to the density of the one-dimensional normal distribution and is given in the next theorem. We leave the proof as an exercise; see Exercise 5. ☞ 59

Theorem 2.3: Density of a Multivariate Random Vector

Let $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the $m \times m$ covariance matrix $\boldsymbol{\Sigma}$ is invertible. Then $\boldsymbol{X}$ has pdf

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^m\,|\boldsymbol{\Sigma}|}}\, \mathrm{e}^{-\frac{1}{2}\,(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \quad \boldsymbol{x} \in \mathbb{R}^m. \tag{2.33}$$

Figure 2.13 shows the pdfs of two bivariate (that is, two-dimensional) normal distributions. In both cases the mean vector is $\boldsymbol{\mu} = [0, 0]^\top$ and the variances (the diagonal elements of $\boldsymbol{\Sigma}$) are 1. The correlation coefficients (or, equivalently here, the covariances) are respectively $\varrho = 0$ and $\varrho = 0.8$.

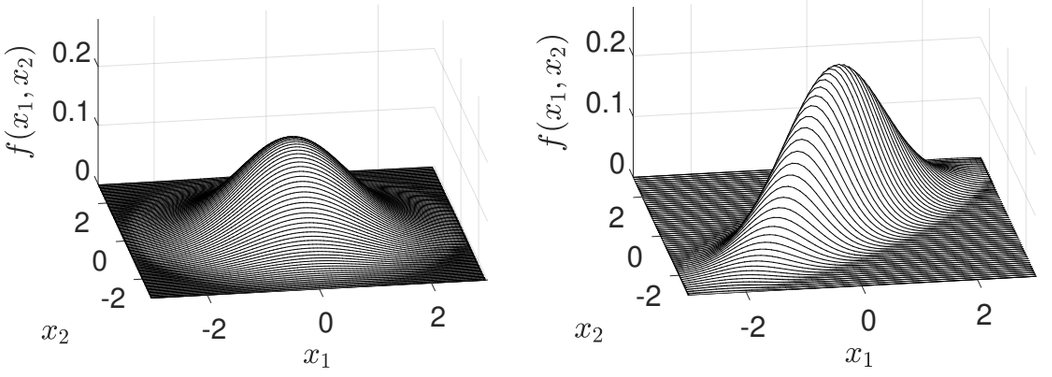

Figure 2.13: Pdfs of bivariate normal distributions with means zero, variances 1, and correlation coefficients 0 (left) and 0.8 (right).

The main reason why the multivariate normal distribution plays an important role in data science and machine learning is that it satisfies the following properties, the details and proofs of which can be found in Appendix C.7:

☞ 436

1. Affine combinations are normal.
2. Marginal distributions are normal.
3. Conditional distributions are normal.

2.8 Normal Linear Models

Normal linear models combine the simplicity of the linear model with the tractability of the Gaussian distribution. They are the principal model for traditional statistics, and include the classic linear regression and analysis of variance models.

Definition 2.3: Normal Linear Model

NORMAL LINEAR MODEL

In a *normal linear model* the response Y depends on a p-dimensional explanatory variable $\boldsymbol{x} = [x_1, \ldots, x_p]^\top$, via the linear relationship

$$Y = \boldsymbol{x}^\top \boldsymbol{\beta} + \varepsilon, \tag{2.34}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Thus, a normal linear model is a linear model (in the sense of Definition 2.1) with normal error terms. Similar to (2.30), the corresponding normal linear model for the whole training set $\{(\boldsymbol{x}_i, Y_i)\}$ has the form

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.35}$$

where $\mathbf{X}$ is the model matrix comprised of rows $\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$. Consequently, $\boldsymbol{Y}$ can be written as $\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \sigma \boldsymbol{Z}$, where $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, so that $\boldsymbol{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$. It follows from (2.33) that its joint density is given by

☞ 45

$$g(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2, \mathbf{X}) = (2\pi\sigma^2)^{-\frac{n}{2}} \, \mathrm{e}^{-\frac{1}{2\sigma^2} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2}. \tag{2.36}$$

Estimation of the parameter $\boldsymbol{\beta}$ can be performed via the least-squares method, as discussed in Example 2.1. An estimate can also be obtained via the maximum likelihood method. This simply means finding the parameters σ^2 and $\boldsymbol{\beta}$ that maximize the likelihood of the outcome $\boldsymbol{y}$, given by the right-hand side of (2.36). It is clear that for every value of σ^2 the likelihood is maximal when $\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ is minimal. As a consequence, the maximum likelihood estimate for $\boldsymbol{\beta}$ is the same as the least-squares estimate (2.15). We leave it as an exercise (see Exercise 18) to show that the maximum likelihood estimate of σ^2 is equal to

☞ 63

$$\widehat{\sigma^2} = \frac{\|\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n}, \tag{2.37}$$

where $\widehat{\boldsymbol{\beta}}$ is the maximum likelihood estimate (least squares estimate in this case) of $\boldsymbol{\beta}$.

2.9 Bayesian Learning

In Bayesian unsupervised learning, we seek to approximate the unknown joint density $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ of the training data $\mathcal{T}_n = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ via a joint pdf of the form

$$\int \left(\prod_{i=1}^{n} g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}) \right) w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}, \tag{2.38}$$

where $g(\cdot \,|\, \boldsymbol{\theta})$ belongs to a family of parametric densities $\mathcal{G}_p := \{g(\cdot \,|\, \boldsymbol{\theta}),\ \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ (viewed as a family of pdfs conditional on a parameter $\boldsymbol{\theta}$ in some set $\boldsymbol{\Theta} \subset \mathbb{R}^p$) and $w(\boldsymbol{\theta})$ is a pdf that belongs to a (possibly different) family of densities $\mathcal{W}_p$. Note how the joint pdf (2.38) satisfies the permutational invariance (2.26) and can thus be useful as a model for training data which is part of an exchangeable sequence of random variables.

Following standard practice in a Bayesian context, instead of writing $f_X(x)$ and $f_{X|Y}(x \,|\, y)$ for the pdf of X and the conditional pdf of X given Y, one simply writes $f(x)$ and $f(x \,|\, y)$. If Y is a different random variable, its pdf (at y) is thus denoted by $f(y)$.

Thus, we will use the same symbol g for different (conditional) approximating probability densities and f for the different (conditional) true and unknown probability densities. Using Bayesian notation, we can write $g(\tau \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{n} g(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})$ and thus the approximating joint pdf (2.38) can then be written as $\int g(\tau \,|\, \boldsymbol{\theta}) \, w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}$ and the true unknown joint pdf as $f(\tau) = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$.

Once $\mathcal{G}_p$ and $\mathcal{W}_p$ are specified, selecting an approximating function $g(\boldsymbol{x})$ of the form

$$g(\boldsymbol{x}) = \int g(\boldsymbol{x} \,|\, \boldsymbol{\theta}) \, w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}$$

is equivalent to selecting a suitable w from $\mathcal{W}_p$. Similar to (2.27), we can use the Kullback–Leibler risk to measure the discrepancy between the proposed approximation (2.38) and the true $f(\tau)$:

$$\ell(g) = \mathbb{E} \ln \frac{f(\mathcal{T})}{\int g(\mathcal{T} \,|\, \boldsymbol{\theta}) \, w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}} = \int f(\tau) \ln \frac{f(\tau)}{\int g(\tau \,|\, \boldsymbol{\theta}) \, w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}} \, \mathrm{d}\tau. \tag{2.39}$$

The main difference with (2.27) is that since the training data is not necessarily iid (it may be exchangeable, for example), the expectation must be with respect to the joint density of $\mathcal{T}$, not with respect to the marginal $f(\boldsymbol{x})$ (as in the iid case). ☞ 40

Minimizing the training loss is equivalent to maximizing the likelihood of the training data τ; that is, solving the optimization problem

$$\max_{w \in \mathcal{W}_p} \int g(\tau \,|\, \boldsymbol{\theta}) \, w(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta},$$

where the maximization is over an appropriate class $\mathcal{W}_p$ of density functions that is believed to result in the smallest KL risk.

Suppose that we have a rough guess, denoted $w_0(\boldsymbol{\theta})$, for the best $w \in \mathcal{W}_p$ that minimizes the Kullback–Leibler risk. We can always increase the resulting likelihood $L_0 := \int g(\tau \,|\, \boldsymbol{\theta})\, w_0(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$ by instead using the density $w_1(\boldsymbol{\theta}) := w_0(\boldsymbol{\theta})\, g(\tau \,|\, \boldsymbol{\theta})/L_0$, giving a likelihood $L_1 := \int g(\tau \,|\, \boldsymbol{\theta})\, w_1(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$. To see this, write L_0 and L_1 as expectations with respect to w_0. In particular, we can write

$$L_0 = \mathbb{E}_{w_0}\, g(\tau \,|\, \boldsymbol{\theta}) \quad \text{and} \quad L_1 = \mathbb{E}_{w_1}\, g(\tau \,|\, \boldsymbol{\theta}) = \mathbb{E}_{w_0} g^2(\tau \,|\, \boldsymbol{\theta})/L_0.$$

It follows that

$$L_1 - L_0 = \frac{1}{L_0}\mathbb{E}_{w_0}\left[g^2(\tau \,|\, \boldsymbol{\theta}) - L_0^2\right] = \frac{1}{L_0}\mathbb{V}\mathrm{ar}_{w_0}[g(\tau \,|\, \boldsymbol{\theta})] \geqslant 0. \tag{2.40}$$

We may thus expect to obtain better predictions using w_1 instead of w_0, because w_1 has taken into account the observed data τ and increased the likelihood of the model. In fact, if we iterate this process (see Exercise 20) and create a sequence of densities $w_1, w_2, \ldots$ such that $w_t(\boldsymbol{\theta}) \propto w_{t-1}(\boldsymbol{\theta})\, g(\tau \,|\, \boldsymbol{\theta})$, then $w_t(\boldsymbol{\theta})$ concentrates more and more of its probability mass at the maximum likelihood estimator $\widehat{\boldsymbol{\theta}}$ (see (2.28)) and in the limit equals a (degenerate) point-mass pdf at $\widehat{\boldsymbol{\theta}}$. In other words, in the limit we recover the maximum likelihood method: $g_\tau(\boldsymbol{x}) = g(\boldsymbol{x} \,|\, \widehat{\boldsymbol{\theta}})$. Thus, unless the class of densities $\mathcal{W}_p$ is restricted to be non-degenerate, maximizing the likelihood as much as possible leads to a degenerate choice for $w(\boldsymbol{\theta})$.

In many situations, the maximum likelihood estimate $g(\tau \,|\, \widehat{\boldsymbol{\theta}})$ is either not an appropriate approximation to $f(\tau)$ (see Example 2.9), or simply fails to exist (see Exercise 10 in Chapter 4). In such cases, given an initial non-degenerate guess $w_0(\boldsymbol{\theta}) = g(\boldsymbol{\theta})$, one can obtain a more appropriate and non-degenerate approximation to $f(\tau)$ by taking $w(\boldsymbol{\theta}) = w_1(\boldsymbol{\theta}) \propto g(\tau \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})$ in (2.38), giving the following Bayesian learner of $f(\boldsymbol{x})$:

☞ 161

$$g_\tau(\boldsymbol{x}) := \int g(\boldsymbol{x} \,|\, \boldsymbol{\theta}) \frac{g(\tau \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})}{\int g(\tau \,|\, \boldsymbol{\vartheta})\, g(\boldsymbol{\vartheta})\, \mathrm{d}\boldsymbol{\vartheta}}\, \mathrm{d}\boldsymbol{\theta}, \tag{2.41}$$

where $\int g(\tau \,|\, \boldsymbol{\vartheta})\, g(\boldsymbol{\vartheta})\, \mathrm{d}\boldsymbol{\vartheta} = g(\tau)$. Using Bayes' formula for probability densities,

☞ 430

$$g(\boldsymbol{\theta} \,|\, \tau) = \frac{g(\tau \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})}{g(\tau)}, \tag{2.42}$$

we can write $w_1(\boldsymbol{\theta}) = g(\boldsymbol{\theta} \,|\, \tau)$. With this notation, we have the following definitions.

Definition 2.4: Prior, Likelihood, and Posterior

Let τ and $\mathcal{G}_p := \{g(\cdot \,|\, \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ be the training set and family of approximating functions.

PRIOR
- A pdf $g(\boldsymbol{\theta})$ that reflects our *a priori* beliefs about $\boldsymbol{\theta}$ is called the *prior* pdf.

LIKELIHOOD
- The conditional pdf $g(\tau \,|\, \boldsymbol{\theta})$ is called the *likelihood*.

POSTERIOR
- Inference about $\boldsymbol{\theta}$ is given by the *posterior* pdf $g(\boldsymbol{\theta} \,|\, \tau)$, which is proportional to the product of the prior and the likelihood:

$$g(\boldsymbol{\theta} \,|\, \tau) \propto g(\tau \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta}).$$

■ **Remark 2.1 (Early Stopping)** Bayes iteration is an example of an "early stopping" heuristic for maximum likelihood optimization, where we exit after only one step. As observed above, if we keep iterating, we obtain the maximum likelihood estimate (MLE). In a sense the Bayes rule provides a regularization of the MLE. Regularization is discussed in more detail in Chapter 6; see also Example 2.9. The early stopping rule is also of benefit in regularization; see Exercise 20 in Chapter 6. ■

On the one hand, the initial guess $g(\boldsymbol{\theta})$ conveys the *a priori* (prior to training the Bayesian learner) information about the optimal density in $\mathcal{W}_p$ that minimizes the KL risk. Using this prior $g(\boldsymbol{\theta})$, the Bayesian approximation to $f(\boldsymbol{x})$ is the *prior predictive density*:

PRIOR PREDICTIVE DENSITY

$$g(\boldsymbol{x}) = \int g(\boldsymbol{x} \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}.$$

On the other hand, the posterior pdf conveys improved knowledge about this optimal density in $\mathcal{W}_p$ after training with τ. Using the posterior $g(\boldsymbol{\theta} \,|\, \tau)$, the Bayesian learner of $f(\boldsymbol{x})$ is the *posterior predictive density*:

POSTERIOR PREDICTIVE DENSITY

$$g_\tau(\boldsymbol{x}) = g(\boldsymbol{x} \,|\, \tau) = \int g(\boldsymbol{x} \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta} \,|\, \tau)\, \mathrm{d}\boldsymbol{\theta},$$

where we have assumed that $g(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \tau) = g(\boldsymbol{x} \,|\, \boldsymbol{\theta})$; that is, the likelihood depends on τ only through the parameter $\boldsymbol{\theta}$.

The choice of the prior is typically governed by two considerations:

1. the prior should be simple enough to facilitate the computation or simulation of the posterior pdf;
2. the prior should be general enough to model ignorance of the parameter of interest.

Priors that do not convey much knowledge of the parameter are said to be *uninformative*. The uniform or *flat* prior in Example 2.9 (to follow) is frequently used.

UNINFORMATIVE PRIOR

For the purpose of analytical and numerical computations, we can view $\boldsymbol{\theta}$ as a random vector with prior density $g(\boldsymbol{\theta})$, which after training is updated to the posterior density $g(\boldsymbol{\theta} \,|\, \tau)$.

The above thinking allows us to write $g(\boldsymbol{x} \,|\, \tau) \propto \int g(\boldsymbol{x} \,|\, \boldsymbol{\theta})\, g(\tau \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$, for example, thus ignoring any constants that do not depend on the argument of the densities.

■ **Example 2.7 (Normal Model)** Suppose that the training data $\mathcal{T} = \{X_1, \ldots, X_n\}$ is modeled using the likelihood $g(x \,|\, \boldsymbol{\theta})$ that is the pdf of

$$X \,|\, \boldsymbol{\theta} \sim \mathcal{N}(\mu, \sigma^2),$$

where $\boldsymbol{\theta} := [\mu, \sigma^2]^\top$. Next, we need to specify the prior distribution of $\boldsymbol{\theta}$ to complete the model. We can specify prior distributions for μ and σ^2 separately and then take their product to obtain the prior for vector $\boldsymbol{\theta}$ (assuming independence). A possible prior distribution for μ is

$$\mu \sim \mathcal{N}(\nu, \phi^2). \tag{2.43}$$

HYPERPARAMETERS

It is typical to refer to any parameters of the prior density as *hyperparameters* of the Bayesian model. Instead of giving directly a prior for σ^2 (or σ), it turns out to be convenient to give the following prior distribution to $1/\sigma^2$:

$$\frac{1}{\sigma^2} \sim \mathsf{Gamma}(\alpha, \beta). \tag{2.44}$$

INVERSE GAMMA

☞ 63

The smaller α and β are, the less informative is the prior. Under this prior, σ^2 is said to have an *inverse gamma*[3] distribution. If $1/Z \sim \mathsf{Gamma}(\alpha, \beta)$, then the pdf of Z is proportional to $\exp(-\beta/z)/z^{\alpha+1}$ (Exercise 19). The Bayesian posterior is then given by:

$$\begin{aligned} g(\mu, \sigma^2 \,|\, \tau) &\propto g(\mu) \times g(\sigma^2) \times g(\tau \,|\, \mu, \sigma^2) \\ &\propto \exp\left\{-\frac{(\mu-\nu)^2}{2\phi^2}\right\} \times \frac{\exp\left\{-\beta/\sigma^2\right\}}{(\sigma^2)^{\alpha+1}} \times \frac{\exp\left\{-\sum_i (x_i-\mu)^2/(2\sigma^2)\right\}}{(\sigma^2)^{n/2}} \\ &\propto (\sigma^2)^{-n/2-\alpha-1} \exp\left\{-\frac{(\mu-\nu)^2}{2\phi^2} - \frac{\beta}{\sigma^2} - \frac{(\mu-\bar{x}_n)^2 + S_n^2}{2\sigma^2/n}\right\}, \end{aligned}$$

where $S_n^2 := \frac{1}{n}\sum_i x_i^2 - \bar{x}_n^2 = \frac{1}{n}\sum_i (x_i - \bar{x}_n)^2$ is the (scaled) sample variance. All inference about (μ, σ^2) is then represented by the posterior pdf. To facilitate computations it is helpful to find out if the posterior belongs to a recognizable family of distributions. For example, the conditional pdf of μ given σ^2 and τ is

$$g(\mu \,|\, \sigma^2, \tau) \propto \exp\left\{-\frac{(\mu-\nu)^2}{2\phi^2} - \frac{(\mu-\bar{x}_n)^2}{2\sigma^2/n}\right\},$$

which after simplification can be recognized as the pdf of

$$(\mu \,|\, \sigma^2, \tau) \sim \mathcal{N}\left(\gamma_n \bar{x}_n + (1-\gamma_n)\nu,\ \gamma_n \sigma^2/n\right), \tag{2.45}$$

where we have defined the weight parameter: $\gamma_n := \frac{n}{\sigma^2} \Big/ \left(\frac{1}{\phi^2} + \frac{n}{\sigma^2}\right)$. We can then see that the posterior mean $\mathbb{E}[\mu \,|\, \sigma^2, \tau] = \gamma_n \bar{x}_n + (1-\gamma_n)\nu$ is a weighted linear combination of the prior mean ν and the sample average $\bar{x}_n$. Further, as $n \to \infty$, the weight $\gamma_n \to 1$ and thus the posterior mean approaches the maximum likelihood estimate $\bar{x}_n$. ■

IMPROPER PRIOR

It is sometimes possible to use a prior $g(\boldsymbol{\theta})$ that is not a *bona fide* probability density, in the sense that $\int g(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \infty$, as long as the resulting posterior $g(\boldsymbol{\theta} \,|\, \tau) \propto g(\tau \,|\, \boldsymbol{\theta}) g(\boldsymbol{\theta})$ is a proper pdf. Such a prior is called an *improper prior*.

■ **Example 2.8 (Normal Model (cont.))** An example of an improper prior is obtained from (2.43) when we let $\phi \to \infty$ (the larger ϕ is, the more uninformative is the prior). Then, $g(\mu) \propto 1$ is a flat prior, but $\int g(\mu)\, \mathrm{d}\mu = \infty$, making it an improper prior. Nevertheless, the posterior is a proper density, and in particular the conditional posterior of $(\mu \,|\, \sigma^2, \tau)$ simplifies to

$$(\mu \,|\, \sigma^2, \tau) \sim \mathcal{N}\left(\bar{x}_n, \sigma^2/n\right),$$

[3] Reciprocal gamma distribution would have been a better name.

because the weight parameter γ_n goes to 1 as $\phi \to \infty$. The improper prior $g(\mu) \propto 1$ also allows us to simplify the posterior marginal for σ^2:

$$g(\sigma^2 \,|\, \tau) = \int g(\mu, \sigma^2 \,|\, \tau)\, \mathrm{d}\mu \propto (\sigma^2)^{-(n-1)/2-\alpha-1} \exp\left\{-\frac{\beta + nS_n^2/2}{\sigma^2}\right\},$$

which we recognize as the density corresponding to

$$\frac{1}{\sigma^2} \,\Big|\, \tau \sim \mathsf{Gamma}\left(\alpha + \frac{n-1}{2},\, \beta + \frac{n}{2} S_n^2\right).$$

In addition to $g(\mu) \propto 1$, we can also use an improper prior for σ^2. If we take the limit $\alpha \to 0$ and $\beta \to 0$ in (2.44), then we also obtain the improper prior $g(\sigma^2) \propto 1/\sigma^2$ (or equivalently $g(1/\sigma^2) \propto 1/\sigma^2$). In this case, the posterior marginal density for σ^2 implies that:

$$\frac{nS_n^2}{\sigma^2} \,\Big|\, \tau \sim \chi^2_{n-1}$$

and the posterior marginal density for μ implies that:

$$\frac{\mu - \bar{x}_n}{S_n/\sqrt{n-1}} \,\Big|\, \tau \sim \mathsf{t}_{n-1}. \tag{2.46}$$

In general, deriving a simple formula for the posterior density of $\boldsymbol{\theta}$ is either impossible or too tedious. Instead, the Monte Carlo methods in Chapter 3 can be used to simulate (approximately) from the posterior for the purposes of inference and prediction. ■

One way in which a distributional result such as (2.46) can be useful is in the construction of a 95% *credible interval* $\mathcal{I}$ for the parameter μ; that is, an interval $\mathcal{I}$ such that the probability $\mathbb{P}[\mu \in \mathcal{I} \,|\, \tau]$ is equal to 0.95. For example, the symmetric 95% credible interval is

CREDIBLE INTERVAL

$$\mathcal{I} = \left[\bar{x}_n - \frac{S_n}{\sqrt{n-1}}\gamma,\ \bar{x}_n + \frac{S_n}{\sqrt{n-1}}\gamma\right],$$

where γ is the 0.975-quantile of the t_{n-1} distribution. Note that the credible interval is not a random object and that the parameter μ is interpreted as a random variable with a distribution. This is unlike the case of classical confidence intervals, where the parameter is nonrandom, but the interval is (the outcome of) a random object. ☞ 459

As a generalization of the 95% Bayesian credible interval we can define a $1-\alpha$ *credible region*, which is any set $\mathcal{R}$ satisfying

CREDIBLE REGION

$$\mathbb{P}[\boldsymbol{\theta} \in \mathcal{R} \,|\, \tau] = \int_{\boldsymbol{\theta} \in \mathcal{R}} g(\boldsymbol{\theta} \,|\, \tau)\, \mathrm{d}\boldsymbol{\theta} \geqslant 1 - \alpha. \tag{2.47}$$

■ **Example 2.9 (Bayesian Regularization of Maximum Likelihood)** Consider modeling the number of deaths during birth in a maternity ward. Suppose that the hospital data consists of $\tau = \{x_1, \ldots, x_n\}$, with $x_i = 1$ if the i-th baby has died during birth and $x_i = 0$ otherwise, for $i = 1, \ldots, n$. A possible Bayesian model for the data is $\theta \sim \mathcal{U}(0,1)$ (uniform prior) with $(X_1, \ldots, X_n \,|\, \theta) \overset{\text{iid}}{\sim} \mathsf{Ber}(\theta)$. The likelihood is therefore

$$g(\tau \,|\, \theta) = \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{1-x_i} = \theta^s \, (1-\theta)^{n-s},$$

where $s = x_1 + \cdots + x_n$ is the total number of deaths. Since $g(\theta) = 1$, the posterior pdf is

$$g(\theta \,|\, \tau) \propto \theta^s \, (1-\theta)^{n-s}, \quad \theta \in [0,1],$$

which is the pdf of the $\mathsf{Beta}(s+1, n-s+1)$ distribution. The normalization constant is $(n+1)\binom{n}{s}$. The posterior pdf is shown in Figure 2.14 for $(s,n) = (0,100)$. It is not difficult

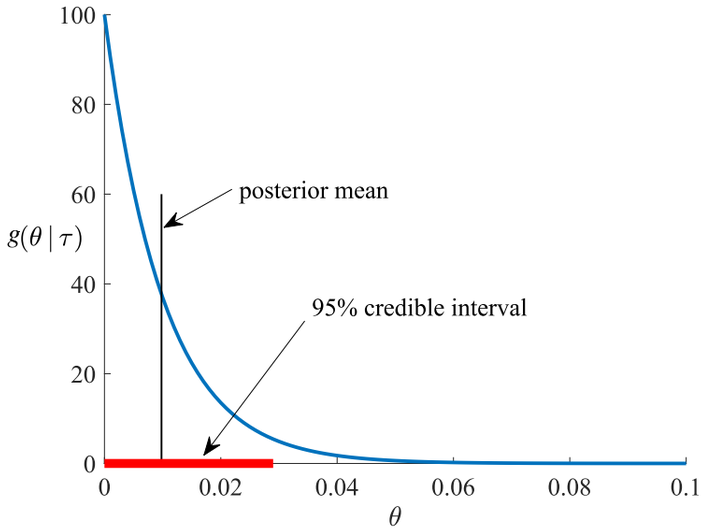

Figure 2.14: Posterior pdf for θ, with $n = 100$ and $s = 0$.

MAXIMUM A POSTERIORI

to see that the *maximum a posteriori* (MAP) estimate of θ (the mode or maximizer of the posterior density) is

$$\underset{\theta}{\text{argmax}}\, g(\theta \,|\, \tau) = \frac{s}{n},$$

which agrees with the maximum likelihood estimate. Figure 2.14 also shows that the left one-sided 95% credible interval for θ is $[0, 0.0292]$, where 0.0292 is the 0.95 quantile (rounded) of the $\mathsf{Beta}(1, 101)$ distribution.

Observe that when $(s,n) = (0,100)$ the maximum likelihood estimate $\widehat{\theta} = 0$ infers that deaths at birth are not possible. We know that this inference is wrong — the probability of death can never be zero, it is simply (and fortunately) too small to be inferred accurately from a sample size of $n = 100$. In contrast to the maximum likelihood estimate, the posterior mean $\mathbb{E}[\theta \,|\, \tau] = (s+1)/(n+2)$ is not zero for $(s,n) = (0,100)$ and provides the more reasonable point estimate of 0.0098 for the probability of death.

In addition, while computing a Bayesian credible interval poses no conceptual difficulties, it is not simple to derive a confidence interval for the maximum likelihood estimate of $\widehat{\theta}$, because the likelihood as a function of θ is not differentiable at $\theta = 0$. As a result of this lack of smoothness, the usual confidence intervals based on the normal approximation cannot be used. ■

We now return to the unsupervised learning setting of Section 2.6, but consider this from a Bayesian perspective. Recall from (2.39) that the Kullback–Leibler risk for an approximating function g is

$$\ell(g) = \int f(\tau'_n)[\ln f(\tau'_n) - \ln g(\tau'_n)]\,\mathrm{d}\tau'_n,$$

where τ'_n denotes the test data. Since $\int f(\tau'_n) \ln f(\tau'_n)\,\mathrm{d}\tau'_n$ plays no role in minimizing the risk, we consider instead the *cross-entropy risk*, defined as ☞ 122

$$\ell(g) = -\int f(\tau'_n) \ln g(\tau'_n)\,\mathrm{d}\tau'_n.$$

Note that the smallest possible cross-entropy risk is $\ell_n^* = -\int f(\tau'_n) \ln f(\tau'_n)\,\mathrm{d}\tau'_n$. The expected generalization risk of the Bayesian learner can then be decomposed as

$$\mathbb{E}\,\ell(g_{\mathcal{T}_n}) = \ell_n^* + \underbrace{\int f(\tau'_n) \ln \frac{f(\tau'_n)}{\mathbb{E}\,g(\tau'_n \,|\, \mathcal{T}_n)}\,\mathrm{d}\tau'_n}_{\text{“bias” component}} + \underbrace{\mathbb{E} \int f(\tau'_n) \ln \frac{\mathbb{E}\,g(\tau'_n \,|\, \mathcal{T}_n)}{g(\tau'_n \,|\, \mathcal{T}_n)}\,\mathrm{d}\tau'_n}_{\text{“variance” component}},$$

where $g_{\mathcal{T}_n}(\tau'_n) = g(\tau'_n \,|\, \mathcal{T}_n) = \int g(\tau'_n \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta} \,|\, \mathcal{T}_n)\,\mathrm{d}\boldsymbol{\theta}$ is the posterior predictive density after observing $\mathcal{T}_n$.

Assuming that the sets $\mathcal{T}_n$ and $\mathcal{T}'_n$ are comprised of $2n$ iid random variables with density f, we can show (Exercise 23) that the expected generalization risk simplifies to

$$\mathbb{E}\,\ell(g_{\mathcal{T}_n}) = \mathbb{E} \ln g(\mathcal{T}_n) - \mathbb{E} \ln g(\mathcal{T}_{2n}), \tag{2.48}$$

where $g(\tau_n)$ and $g(\tau_{2n})$ are the prior predictive densities of τ_n and τ_{2n}, respectively.

Let $\overline{\boldsymbol{\theta}}_n = \operatorname{argmax}_{\boldsymbol{\theta}} g(\boldsymbol{\theta} \,|\, \mathcal{T}_n)$ be the MAP estimator of $\boldsymbol{\theta}^* := \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E} \ln g(\boldsymbol{X} \,|\, \boldsymbol{\theta})$. Assuming that $\overline{\boldsymbol{\theta}}_n$ converges to $\boldsymbol{\theta}^*$ (with probability one) and $\frac{1}{n}\mathbb{E} \ln g(\mathcal{T}_n \,|\, \overline{\boldsymbol{\theta}}_n) = \mathbb{E} \ln g(\boldsymbol{X} \,|\, \boldsymbol{\theta}^*) + O(1/n)$, we can use the following large-sample approximation of the expected generalization risk.

Theorem 2.4: Approximating the Bayesian Cross-Entropy Risk

For $n \to \infty$, the expected cross-entropy generalization risk satisfies:

$$\mathbb{E}\ell(g_{\mathcal{T}_n}) \simeq -\mathbb{E} \ln g(\mathcal{T}_n) - \frac{p}{2} \ln n, \tag{2.49}$$

where (with p the dimension of the parameter vector $\boldsymbol{\theta}$ and $\overline{\boldsymbol{\theta}}_n$ the MAP estimator):

$$\mathbb{E} \ln g(\mathcal{T}_n) \simeq \mathbb{E} \ln g(\mathcal{T}_n \,|\, \overline{\boldsymbol{\theta}}_n) - \frac{p}{2} \ln n. \tag{2.50}$$

☞ 452

Proof: To show (2.50), we apply Theorem C.21 to $\ln \int \mathrm{e}^{-nr_n(\boldsymbol{\theta})} g(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}$, where

$$r_n(\boldsymbol{\theta}) := -\frac{1}{n} \ln g(\mathcal{T}_n \,|\, \boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \ln g(\boldsymbol{X}_i \,|\, \boldsymbol{\theta}) \xrightarrow{\text{a.s.}} -\mathbb{E} \ln g(\boldsymbol{X} \,|\, \boldsymbol{\theta}) =: r(\boldsymbol{\theta}) < \infty.$$

This gives (with probability one)

$$\ln \int g(\mathcal{T}_n \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \simeq -nr(\boldsymbol{\theta}^*) - \frac{p}{2} \ln(n).$$

Taking expectations on both sides and using $nr(\boldsymbol{\theta}^*) = n\mathbb{E}[r_n(\overline{\boldsymbol{\theta}}_n)] + O(1)$, we deduce (2.50). To demonstrate (2.49), we derive the asymptotic approximation of $\mathbb{E} \ln g(\mathcal{T}_{2n})$ by repeating the argument for (2.50), but replacing n with $2n$, where necessary. Thus, we obtain:

$$\mathbb{E} \ln g(\mathcal{T}_{2n}) \simeq -2nr(\boldsymbol{\theta}^*) - \frac{p}{2} \ln(2n).$$

Then, (2.49) follows from the identity (2.48). □

The results of Theorem 2.4 have two major implications for model selection and assessment. First, (2.49) suggests that $-\ln g(\mathcal{T}_n)$ can be used as a crude (leading-order) asymptotic approximation to the expected generalization risk for large n and fixed p. In this context, the prior predictive density $g(\mathcal{T}_n)$ is usually called the *model evidence* or *marginal likelihood* for the class $\mathcal{G}_p$. Since the integral $\int g(\mathcal{T}_n \,|\, \boldsymbol{\theta})\, g(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}$ is rarely available in closed form, the exact computation of the model evidence is typically not feasible and may require Monte Carlo estimation methods.

MODEL EVIDENCE

☞ 78

Second, when the model evidence is difficult to compute via Monte Carlo methods or otherwise, (2.50) suggests that we can use the following large-sample approximation:

$$-2\mathbb{E} \ln g(\mathcal{T}_n) \simeq -2 \ln g(\mathcal{T}_n \,|\, \overline{\boldsymbol{\theta}}_n) + p \ln(n). \tag{2.51}$$

The asymptotic approximation on the right-hand side of (2.51) is called the *Bayesian information criterion* (BIC). We prefer the class $\mathcal{G}_p$ with the smallest BIC. The BIC is typically used when the model evidence is difficult to compute and n is sufficiently larger than p. For a fixed p, and as n becomes larger and larger, the BIC becomes a more and more accurate estimator of $-2\mathbb{E} \ln g(\mathcal{T}_n)$. Note that the BIC approximation is valid even when the true density $f \notin \mathcal{G}_p$. The BIC provides an alternative to the *Akaike information criterion* (AIC) for model selection. However, while the BIC approximation does not assume that the true model f belongs to the parametric class under consideration, the AIC assumes that $f \in \mathcal{G}_p$. Thus, the AIC is merely a *heuristic* approximation based on the asymptotic approximations in Theorem 4.1.

BAYESIAN INFORMATION CRITERION

☞ 126

Although the above Bayesian theory has been presented in an unsupervised learning setting, it can be readily extended to the supervised case. We only need to relabel the training set $\mathcal{T}_n$. In particular, when (as is typical for regression models) the training responses $Y_1, \ldots, Y_n$ are considered as random variables but the corresponding feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are viewed as being fixed, then $\mathcal{T}_n$ is the collection of random responses $\{Y_1, \ldots, Y_n\}$. Alternatively, we can simply identify $\mathcal{T}_n$ with the response vector $\boldsymbol{Y} = [Y_1, \ldots, Y_n]^\top$. We will adopt this notation in the next example.

■ **Example 2.10 (Polynomial Regression (cont.))** Consider Example 2.2 once again, but now in a Bayesian framework, where the prior knowledge on $(\sigma^2, \boldsymbol{\beta})$ is specified by $g(\sigma^2) = 1/\sigma^2$ and $\boldsymbol{\beta} \,|\, \sigma^2 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{D})$, and $\mathbf{D}$ is a (matrix) hyperparameter. Let $\boldsymbol{\Sigma} := (\mathbf{X}^\top \mathbf{X} + \mathbf{D}^{-1})^{-1}$. Then the posterior can be written as:

$$\begin{aligned} g(\boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{y}) &= \frac{\exp\left(-\frac{\|\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}\|^2}{2\sigma^2}\right)}{(2\pi\sigma^2)^{n/2}} \times \frac{\exp\left(-\frac{\boldsymbol{\beta}^\top \mathbf{D}^{-1}\boldsymbol{\beta}}{2\sigma^2}\right)}{(2\pi\sigma^2)^{p/2}\,|\mathbf{D}|^{1/2}} \times \frac{1}{\sigma^2} \Big/ g(\boldsymbol{y}) \\ &= \frac{(\sigma^2)^{-(n+p)/2-1}}{(2\pi)^{(n+p)/2}\,|\mathbf{D}|^{1/2}} \exp\left(-\frac{\|\Sigma^{-1/2}(\boldsymbol{\beta}-\overline{\boldsymbol{\beta}})\|^2}{2\sigma^2} - \frac{(n+p+2)\,\overline{\sigma}^2}{2\sigma^2}\right) \Big/ g(\boldsymbol{y}), \end{aligned}$$

where $\overline{\boldsymbol{\beta}} := \boldsymbol{\Sigma}\mathbf{X}^\top \boldsymbol{y}$ and $\overline{\sigma}^2 := \boldsymbol{y}^\top(\mathbf{I} - \mathbf{X}\boldsymbol{\Sigma}\mathbf{X}^\top)\boldsymbol{y}/(n+p+2)$ are the MAP estimates of $\boldsymbol{\beta}$ and σ^2, and $g(\boldsymbol{y})$ is the model evidence for $\mathcal{G}_p$:

$$\begin{aligned} g(\boldsymbol{y}) &= \iint g(\boldsymbol{\beta}, \sigma^2, \boldsymbol{y})\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\sigma^2 \\ &= \frac{|\boldsymbol{\Sigma}|^{1/2}}{(2\pi)^{n/2}|\mathbf{D}|^{1/2}} \int_0^\infty \frac{\exp\left(-\frac{(n+p+2)\,\overline{\sigma}^2}{2\sigma^2}\right)}{(\sigma^2)^{n/2+1}}\, \mathrm{d}\sigma^2 \\ &= \frac{|\boldsymbol{\Sigma}|^{1/2}\Gamma(n/2)}{|\mathbf{D}|^{1/2}(\pi(n+p+2)\,\overline{\sigma}^2)^{n/2}}. \end{aligned}$$

Therefore, based on (2.49), we have

$$2\mathbb{E}\ell(g_{\mathcal{T}_n}) \simeq -2\ln g(\boldsymbol{y}) = n\ln\left[\pi(n+p+2)\,\overline{\sigma}^2\right] - 2\ln\Gamma(n/2) + \ln|\mathbf{D}| - \ln|\boldsymbol{\Sigma}|.$$

On the other hand, the minus of the log-likelihood of $\boldsymbol{Y}$ can be written as

$$\begin{aligned} -\ln g(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) &= \frac{\|\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}\|^2}{2\sigma^2} + \frac{n}{2}\ln(2\pi\sigma^2) \\ &= \frac{\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}-\overline{\boldsymbol{\beta}})\|^2}{2\sigma^2} + \frac{(n+p+2)\,\overline{\sigma}^2}{2\sigma^2} + \frac{n}{2}\ln(2\pi\sigma^2). \end{aligned}$$

Therefore, the BIC approximation (2.51) is

$$-2\ln g(\boldsymbol{y} \,|\, \overline{\boldsymbol{\beta}}, \overline{\sigma}^2) + (p+1)\ln(n) = n[\ln(2\pi\,\overline{\sigma}^2) + 1] + (p+1)\ln(n) + (p+2), \qquad (2.52)$$

where the extra $\ln(n)$ term in $(p+1)\ln(n)$ is due to the inclusion of σ^2 in $\boldsymbol{\theta} = (\sigma^2, \boldsymbol{\beta})$. Figure 2.15 shows the model evidence and its BIC approximation, where we used a hyperparameter $\mathbf{D} = 10^4 \times \mathbf{I}_p$ for the prior density of $\boldsymbol{\beta}$. We can see that both approximations exhibit a pronounced minimum at $p = 4$, thus identifying the true polynomial regression model. Compare the overall qualitative shape of the cross-entropy risk estimate with the shape of the square-error risk estimate in Figure 2.11.

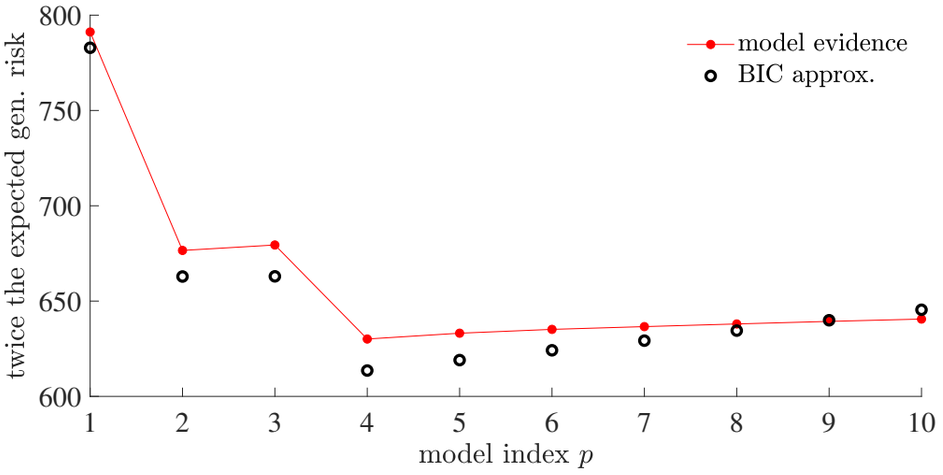

Figure 2.15: The BIC and marginal likelihood used for model selection.

■

It is possible to give the model complexity parameter p a Bayesian treatment, in which we define a prior density on the set of all models under consideration. For example, let $g(p)$, $p = 1, \ldots, m$ be a prior density on m candidate models. Treating the model complexity index p as an additional parameter to $\boldsymbol{\theta} \in \mathbb{R}^p$, and applying Bayes' formula, the posterior for $(\boldsymbol{\theta}, p)$ can be written as:

$$\begin{aligned} g(\boldsymbol{\theta}, p \,|\, \tau) &= g(\boldsymbol{\theta} \,|\, p, \tau) \times g(p \,|\, \tau) \\ &= \underbrace{\frac{g(\tau \,|\, \boldsymbol{\theta}, p)\, g(\boldsymbol{\theta} \,|\, p)}{g(\tau \,|\, p)}}_{\text{posterior of } \boldsymbol{\theta} \text{ given model } p} \times \underbrace{\frac{g(\tau \,|\, p)\, g(p)}{g(\tau)}}_{\text{posterior of model } p}. \end{aligned}$$

The model evidence for a fixed p is now interpreted as the prior predictive density of τ, conditional on the model p:

$$g(\tau \,|\, p) = \int g(\tau \,|\, \boldsymbol{\theta}, p)\, g(\boldsymbol{\theta} \,|\, p)\, \mathrm{d}\boldsymbol{\theta},$$

and the quantity $g(\tau) = \sum_{p=1}^{m} g(\tau \,|\, p)\, g(p)$ is interpreted as the marginal likelihood of all the m candidate models. Finally, a simple method for model selection is to pick the index $\widehat{p}$ with the largest posterior probability:

$$\widehat{p} = \operatorname*{argmax}_p g(p \,|\, \tau) = \operatorname*{argmax}_p g(\tau \,|\, p)\, g(p).$$

■ **Example 2.11 (Polynomial Regression (cont.))** Let us revisit Example 2.10 by giving the parameter $p = 1, \ldots, m$, with $m = 10$, a Bayesian treatment. Recall that we used the notation $\tau = \boldsymbol{y}$ in that example. We assume that the prior $g(p) = 1/m$ is flat and uninformative so that the posterior is given by

$$g(p \,|\, \boldsymbol{y}) \propto g(\boldsymbol{y} \,|\, p) = \frac{|\boldsymbol{\Sigma}|^{1/2}\, \Gamma(n/2)}{|\mathbf{D}|^{1/2} (\pi(n + p + 2)\, \overline{\sigma}^2)^{n/2}},$$

where all quantities in $g(\mathbf{y} \mid p)$ are computed using the first p columns of $\mathbf{X}$. Figure 2.16 shows the resulting posterior density $g(p \mid \mathbf{y})$. The figure also shows the posterior density $\widehat{g}(\mathbf{y} \mid p) / \sum_{p=1}^{10} \widehat{g}(\mathbf{y} \mid p)$, where

$$\widehat{g}(\mathbf{y} \mid p) := \exp\left(-\frac{n[\ln(2\pi\,\overline{\sigma}^2) + 1] + (p+1)\ln(n) + (p+2)}{2}\right)$$

is derived from the BIC approximation (2.52). In both cases, there is a clear maximum at $p = 4$, suggesting that a third-degree polynomial is the most appropriate model for the data.

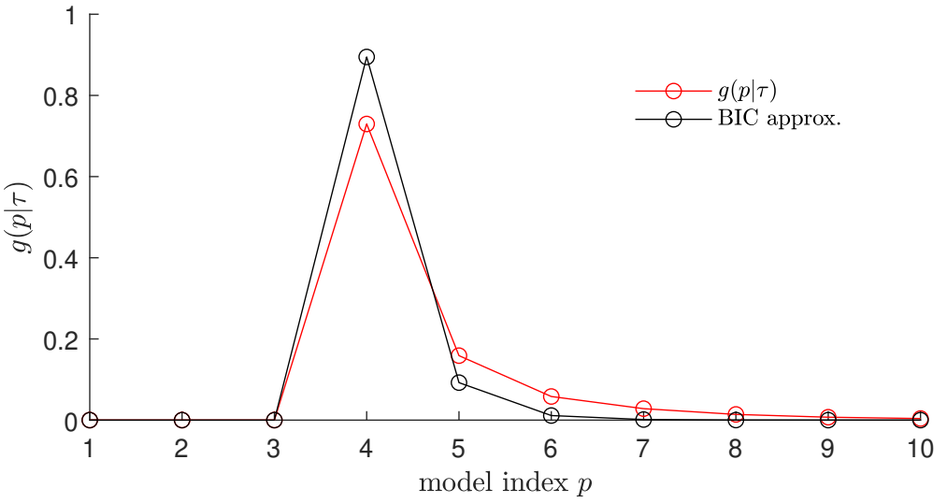

Figure 2.16: Posterior probabilities for each polynomial model of degree $p - 1$.

■

Suppose that we wish to compare two models, say model $p = 1$ and model $p = 2$. Instead of computing the posterior $g(p \mid \tau)$ explicitly, we can compare the posterior odds ratio:

$$\frac{g(p=1 \mid \tau)}{g(p=2 \mid \tau)} = \frac{g(p=1)}{g(p=2)} \times \underbrace{\frac{g(\tau \mid p=1)}{g(\tau \mid p=2)}}_{\text{Bayes factor } B_{1\mid 2}}.$$

This gives rise to the *Bayes factor* $B_{i\mid j}$, whose value signifies the strength of the evidence in favor of model i over model j. In particular $B_{i\mid j} > 1$ means that the evidence in favor for model i is larger. BAYES FACTOR

■ **Example 2.12 (Savage–Dickey Ratio)** Suppose that we have two models. Model $p = 2$ has a likelihood $g(\tau \mid \mu, \nu, p = 2)$, depending on two parameters. Model $p = 1$ has the same functional form for the likelihood but now ν is fixed to some (known) ν_0; that is, $g(\tau \mid \mu, p = 1) = g(\tau \mid \mu, \nu = \nu_0, p = 2)$. We also assume that the prior information on μ

for model 1 is the same as that for model 2, conditioned on $\nu = \nu_0$. That is, we assume $g(\mu \,|\, p = 1) = g(\mu \,|\, \nu = \nu_0, p = 2)$. As model 2 contains model 1 as a special case, the latter is said to be *nested* inside model 2. We can formally write (see also Exercise 26):

$$\begin{aligned} g(\tau \,|\, p = 1) &= \int g(\tau \,|\, \mu, p = 1)\, g(\mu \,|\, p = 1)\, \mathrm{d}\mu \\ &= \int g(\tau \,|\, \mu, \nu = \nu_0, p = 2)\, g(\mu \,|\, \nu = \nu_0, p = 2)\, \mathrm{d}\mu \\ &= g(\tau \,|\, \nu = \nu_0, p = 2) = \frac{g(\tau, \nu = \nu_0 \,|\, p = 2)}{g(\nu = \nu_0 \,|\, p = 2)}. \end{aligned}$$

Hence, the Bayes factor simplifies to

$$B_{1|2} = \frac{g(\tau \,|\, p = 1)}{g(\tau \,|\, p = 2)} = \frac{g(\tau, \nu = \nu_0 \,|\, p = 2)}{g(\nu = \nu_0 \,|\, p = 2)} \Big/ g(\tau \,|\, p = 2) = \frac{g(\nu = \nu_0 \,|\, \tau, p = 2)}{g(\nu = \nu_0 \,|\, p = 2)}.$$

In other words, $B_{1|2}$ is the ratio of the posterior density to the prior density of ν, evaluated at $\nu = \nu_0$ and both under the unrestricted model $p = 2$. This ratio of posterior to prior densities is called the *Savage–Dickey density ratio*. ■

SAVAGE–DICKEY DENSITY RATIO

Whether to use a classical (frequentist) or Bayesian model is largely a question of convenience. Classical inference is useful because it comes with a huge repository of ready-to-use results, and requires no (subjective) prior information on the parameters. Bayesian models are useful because the whole theory is based on the elegant Bayes' formula, and uncertainty in the inference (e.g., confidence intervals) can be quantified much more naturally (e.g., credible intervals). A usual practice is to "Bayesify" a classical model, simply by adding some prior information on the parameters.

Further Reading

A popular textbook on statistical learning is [55]. Accessible treatments of mathematical statistics can be found, for example, in [69], [74], and [124]. More advanced treatments are given in [10], [25], and [78]. A good overview of modern-day statistical inference is given in [36]. Classical references on pattern classification and machine learning are [12] and [35]. For advanced learning theory including information theory and Rademacher complexity, we refer to [28] and [109]. An applied reference for Bayesian inference is [46]. For a survey of numerical techniques relevant to computational statistics, see [90].

Exercises

1. Suppose that the loss function is the piecewise linear function

$$\mathrm{Loss}(y, \widehat{y}) = \alpha\, (\widehat{y} - y)_+ + \beta\, (y - \widehat{y})_+, \quad \alpha, \beta > 0,$$

where c_+ is equal to c if $c > 0$, and zero otherwise. Show that the minimizer of the risk $\ell(g) = \mathbb{E}\,\mathrm{Loss}(Y, g(\boldsymbol{X}))$ satisfies

$$\mathbb{P}[Y < g^*(\boldsymbol{x}) \,|\, \boldsymbol{X} = \boldsymbol{x}] = \frac{\beta}{\alpha + \beta}.$$

In other words, $g^*(\boldsymbol{x})$ is the $\beta/(\alpha + \beta)$ quantile of Y, conditional on $\boldsymbol{X} = \boldsymbol{x}$.

2. Show that, for the squared-error loss, the approximation error $\ell(g^{\mathcal{G}}) - \ell(g^*)$ in (2.16), is equal to $\mathbb{E}(g^{\mathcal{G}}(\boldsymbol{X}) - g^*(\boldsymbol{X}))^2$. [Hint: expand $\ell(g^{\mathcal{G}}) = \mathbb{E}(Y - g^*(\boldsymbol{X}) + g^*(\boldsymbol{X}) - g^{\mathcal{G}}(\boldsymbol{X}))^2$.]

3. Suppose $\mathcal{G}$ is the class of *linear* functions. A linear function evaluated at a feature $\boldsymbol{x}$ can be described as $g(\boldsymbol{x}) = \boldsymbol{\beta}^\top \boldsymbol{x}$ for some parameter vector $\boldsymbol{\beta}$ of appropriate dimension. Denote $g^{\mathcal{G}}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}^{\mathcal{G}}$ and $g_\tau^{\mathcal{G}}(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}$. Show that

$$\mathbb{E}\left(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - g^*(\boldsymbol{X})\right)^2 = \mathbb{E}\left(\boldsymbol{X}^\top \widehat{\boldsymbol{\beta}} - \boldsymbol{X}^\top \boldsymbol{\beta}^{\mathcal{G}}\right)^2 + \mathbb{E}\left(\boldsymbol{X}^\top \boldsymbol{\beta}^{\mathcal{G}} - g^*(\boldsymbol{X})\right)^2.$$

Hence, deduce that the statistical error in (2.16) is $\ell(g_\tau^{\mathcal{G}}) - \ell(g^{\mathcal{G}}) = \mathbb{E}\,(g_\tau^{\mathcal{G}}(\boldsymbol{X}) - g^{\mathcal{G}}(\boldsymbol{X}))^2$.

4. Show that formula (2.24) holds for the 0–1 loss with 0–1 response.

5. Let $\boldsymbol{X}$ be an n-dimensional normal random vector with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, where the determinant of $\boldsymbol{\Sigma}$ is non-zero. Show that $\boldsymbol{X}$ has joint probability density

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n\,|\boldsymbol{\Sigma}|}}\, \mathrm{e}^{-\frac{1}{2}\,(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \quad \boldsymbol{x} \in \mathbb{R}^n.$$

6. Let $\widehat{\boldsymbol{\beta}} = \mathbf{A}^+ \boldsymbol{y}$. Using the defining properties of the pseudo-inverse, show that for any $\boldsymbol{\beta} \in \mathbb{R}^p$, ☞ 362

$$\|\mathbf{A}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\| \leqslant \|\mathbf{A}\boldsymbol{\beta} - \boldsymbol{y}\|.$$

7. Suppose that in the polynomial regression Example 2.1 we select the linear class of functions $\mathcal{G}_p$ with $p \geqslant 4$. Then, $g^* \in \mathcal{G}_p$ and the approximation error is zero, because $g^{\mathcal{G}_p}(\boldsymbol{x}) = g^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}$, where $\boldsymbol{\beta} = [10, -140, 400, -250, 0, \ldots, 0]^\top \in \mathbb{R}^p$. Use the tower property to show that the learner $g_\tau(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}$ with $\widehat{\boldsymbol{\beta}} = \mathbf{X}^+ \boldsymbol{y}$, assuming $\mathrm{rank}(\mathbf{X}) \geqslant 4$, is *unbiased*: ☞ 433 UNBIASED

$$\mathbb{E}\, g_{\mathcal{T}}(\boldsymbol{x}) = g^*(\boldsymbol{x}).$$

8. (Exercise 7 continued.) Observe that the learner $g_{\mathcal{T}}$ can be written as a linear combination of the response variable: $g_{\mathcal{T}}(\boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{X}^+ \boldsymbol{Y}$. Prove that for any learner of the form $\boldsymbol{x}^\top \mathbf{A}\boldsymbol{y}$, where $\mathbf{A} \in \mathbb{R}^{p \times n}$ is some matrix and that satisfies $\mathbb{E}_{\mathbf{X}}[\boldsymbol{x}^\top \mathbf{A}\boldsymbol{Y}] = g^*(\boldsymbol{x})$, we have

$$\mathbb{V}\mathrm{ar}_{\mathbf{X}}[\boldsymbol{x}^\top \mathbf{X}^+ \boldsymbol{Y}] \leqslant \mathbb{V}\mathrm{ar}_{\mathbf{X}}[\boldsymbol{x}^\top \mathbf{A}\boldsymbol{Y}],$$

where the equality is achieved for $\mathbf{A} = \mathbf{X}^+$. This is called the *Gauss–Markov inequality*. Hence, using the Gauss–Markov inequality deduce that for the unconditional variance: GAUSS–MARKOV INEQUALITY

$$\mathbb{V}\mathrm{ar}\, g_{\mathcal{T}}(\boldsymbol{x}) \leqslant \mathbb{V}\mathrm{ar}[\boldsymbol{x}^\top \mathbf{A}\boldsymbol{Y}].$$

Deduce that $\mathbf{A} = \mathbf{X}^+$ also minimizes the expected generalization risk.

9. Consider again the polynomial regression Example 2.1. Use the fact that $\mathbb{E}_{\mathbf{X}}\, \widehat{\boldsymbol{\beta}} = \mathbf{X}^+ \boldsymbol{h}^*(\boldsymbol{u})$, where $\boldsymbol{h}^*(\boldsymbol{u}) = \mathbb{E}[\boldsymbol{Y} \,|\, \boldsymbol{U} = \boldsymbol{u}] = [h^*(u_1), \ldots, h^*(u_n)]^\top$, to show that the expected in-sample risk is:

$$\mathbb{E}_{\mathbf{X}}\, \ell_{\mathrm{in}}(g_{\mathcal{T}}) = \ell^* + \frac{\|\boldsymbol{h}^*(\boldsymbol{u})\|^2 - \|\mathbf{X}\mathbf{X}^+ \boldsymbol{h}^*(\boldsymbol{u})\|^2}{n} + \frac{\ell^* p}{n}.$$

Also, use Theorem C.2 to show that the expected statistical error is: ☞ 432

$$\mathbb{E}_{\mathbf{X}}\, (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top \mathbf{H}_p (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \ell^* \mathrm{tr}(\mathbf{X}^+ (\mathbf{X}^+)^\top \mathbf{H}_p) + (\mathbf{X}^+ \boldsymbol{h}^*(\boldsymbol{u}) - \boldsymbol{\beta})^\top \mathbf{H}_p (\mathbf{X}^+ \boldsymbol{h}^*(\boldsymbol{u}) - \boldsymbol{\beta}).$$

10. Consider the setting of the polynomial regression in Example 2.2. Use Theorem C.19 to prove that

☞ 451

$$\sqrt{n}\,(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_p) \xrightarrow{\mathrm{d}} \mathcal{N}\Big(\mathbf{0},\, \ell^* \mathbf{H}_p^{-1} + \mathbf{H}_p^{-1}\mathbf{M}_p\mathbf{H}_p^{-1}\Big), \tag{2.53}$$

where $\mathbf{M}_p := \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top (g^*(\boldsymbol{X}) - g^{\mathcal{G}_p}(\boldsymbol{X}))^2]$ is the matrix with (i,j)-th entry:

$$\int_0^1 u^{i+j-2} (h^{\mathcal{H}_p}(u) - h^*(u))^2 \,\mathrm{d}u,$$

INVERSE HILBERT MATRIX

and $\mathbf{H}_p^{-1}$ is the $p \times p$ *inverse Hilbert matrix* with (i,j)-th entry:

$$(-1)^{i+j}(i+j-1)\binom{p+i-1}{p-j}\binom{p+j-1}{p-i}\binom{i+j-2}{i-1}^2.$$

Observe that $\mathbf{M}_p = \mathbf{0}$ for $p \geqslant 4$, so that the matrix $\mathbf{M}_p$ term is due to choosing a restrictive class $\mathcal{G}_p$ that does not contain the true prediction function.

11. In Example 2.2 we saw that the statistical error can be expressed (see (2.20)) as

$$\int_0^1 \Big([1, \ldots, u^{p-1}](\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p)\Big)^2 \,\mathrm{d}u = (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p)^\top \mathbf{H}_p (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p).$$

By Exercise 10 the random vector $\boldsymbol{Z}_n := \sqrt{n}(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_p)$ has asymptotically a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{V} := \ell^* \mathbf{H}_p^{-1} + \mathbf{H}_p^{-1}\mathbf{M}_p\mathbf{H}_p^{-1}$. Use Theorem C.2 to show that the *expected* statistical error is asymptotically

☞ 432

$$\mathbb{E}\,(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p)^\top \mathbf{H}_p (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_p) \simeq \frac{\ell^* p}{n} + \frac{\mathrm{tr}(\mathbf{M}_p \mathbf{H}_p^{-1})}{n}, \quad n \to \infty. \tag{2.54}$$

Plot this large-sample approximation of the expected statistical error and compare it with the outcome of the statistical error.

We note a subtle technical detail: In general, convergence in distribution does not imply convergence in L_p-norm (see Example C.6), and so here we have implicitly assumed that $\|\boldsymbol{Z}_n\| \xrightarrow{\mathrm{d}} \mathsf{Dist.} \Rightarrow \|\boldsymbol{Z}_n\| \xrightarrow{L_2} \text{constant} := \lim_{n\uparrow\infty} \mathbb{E}\|\boldsymbol{Z}_n\|$.

☞ 444

12. Consider again Example 2.2. The result in (2.53) suggests that $\mathbb{E}\widehat{\boldsymbol{\beta}} \to \boldsymbol{\beta}_p$ as $n \to \infty$, where $\boldsymbol{\beta}_p$ is the solution in the class $\mathcal{G}_p$ given in (2.18). Thus, the large-sample approximation of the pointwise bias of the learner $g_{\mathcal{T}}^{\mathcal{G}_p}(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}$ at $\boldsymbol{x} = [1, \ldots, u^{p-1}]^\top$ is

$$\mathbb{E}\, g_{\mathcal{T}}^{\mathcal{G}_p}(\boldsymbol{x}) - g^*(\boldsymbol{x}) \simeq [1, \ldots, u^{p-1}]\boldsymbol{\beta}_p - [1, u, u^2, u^3]\boldsymbol{\beta}^*, \quad n \to \infty.$$

Use Python to reproduce Figure 2.17, which shows the (large-sample) pointwise squared bias of the learner for $p \in \{1, 2, 3\}$. Note how the bias is larger near the endpoints $u = 0$ and $u = 1$. Explain why the areas under the curves correspond to the approximation errors.

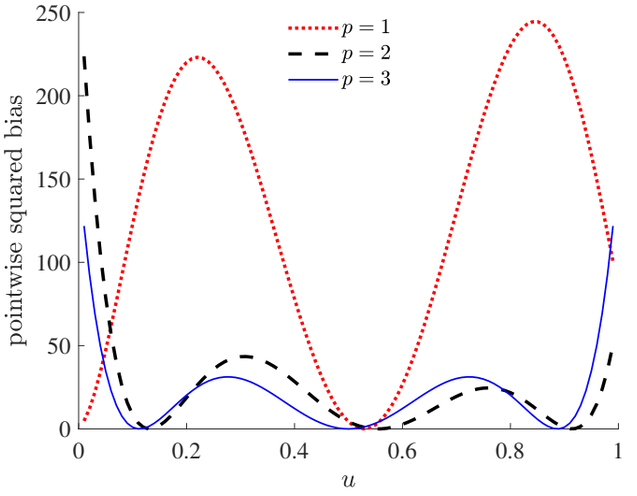

Figure 2.17: The large-sample pointwise squared bias of the learner for $p = 1, 2, 3$. The bias is zero for $p \geqslant 4$.

13. For our running Example 2.2 we can use (2.53) to derive a large-sample approximation of the pointwise variance of the learner $g_{\mathcal{T}}(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}_n$. In particular, show that for large n

$$\mathbb{V}\mathrm{ar}\, g_{\mathcal{T}}(\boldsymbol{x}) \simeq \frac{\ell^* \, \boldsymbol{x}^\top \mathbf{H}_p^{-1} \boldsymbol{x}}{n} + \frac{\boldsymbol{x}^\top \mathbf{H}_p^{-1} \mathbf{M}_p \mathbf{H}_p^{-1} \boldsymbol{x}}{n}, \quad n \to \infty. \tag{2.55}$$

Figure 2.18 shows this (large-sample) variance of the learner for different values of the predictor u and model index p. Observe that the variance ultimately increases in p and that it is smaller at $u = 1/2$ than closer to the endpoints $u = 0$ or $u = 1$. Since the bias is also

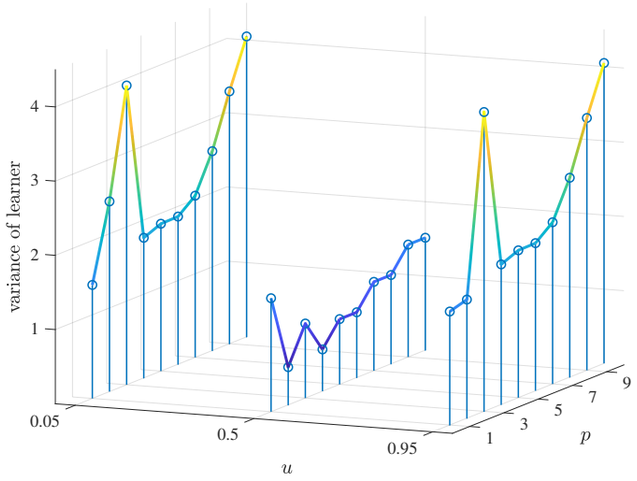

Figure 2.18: The pointwise variance of the learner for various pairs of p and u.

larger near the endpoints, we deduce that the pointwise mean squared error (2.21) is larger near the endpoints of the interval $[0, 1]$ than near its middle. In other words, the error is much smaller in the center of the data cloud than near its periphery.

☞ 405

JENSEN'S INEQUALITY

14. Let $h : \boldsymbol{x} \mapsto \mathbb{R}$ be a convex function and let $\boldsymbol{X}$ be a random variable. Use the subgradient definition of convexity to prove *Jensen's inequality*:

$$\mathbb{E}\, h(\boldsymbol{X}) \geqslant h(\mathbb{E}\boldsymbol{X}). \tag{2.56}$$

15. Using Jensen's inequality, show that the Kullback–Leibler divergence between probability densities f and g is always positive; that is,

$$\mathbb{E} \ln \frac{f(\boldsymbol{X})}{g(\boldsymbol{X})} \geqslant 0,$$

where $\boldsymbol{X} \sim f$.

VAPNIK–CHERNOVENKIS BOUND

16. The purpose of this exercise is to prove the following *Vapnik–Chernovenkis bound*: for any *finite* class $\mathcal{G}$ (containing only a finite number $|\mathcal{G}|$ of possible functions) and a general *bounded* loss function, $l \leqslant \text{Loss} \leqslant u$, the expected statistical error is bounded from above according to:

$$\mathbb{E}\, \ell(g^{\mathcal{G}}_{\mathcal{T}_n}) - \ell(g^{\mathcal{G}}) \leqslant \frac{(u-l)\sqrt{2\ln(2|\mathcal{G}|)}}{\sqrt{n}}. \tag{2.57}$$

Note how this bound conveniently does not depend on the distribution of the training set $\mathcal{T}_n$ (which is typically unknown), but only on the complexity (i.e., cardinality) of the class $\mathcal{G}$. We can break up the proof of (2.57) into the following four parts:

(a) For a general function class $\mathcal{G}$, training set $\mathcal{T}$, risk function ℓ, and training loss $\ell_{\mathcal{T}}$, we have, by definition, $\ell(g^{\mathcal{G}}) \leqslant \ell(g)$ and $\ell_{\mathcal{T}}(g^{\mathcal{G}}_{\mathcal{T}}) \leqslant \ell_{\mathcal{T}}(g)$ for all $g \in \mathcal{G}$. Show that

$$\ell(g^{\mathcal{G}}_{\mathcal{T}}) - \ell(g^{\mathcal{G}}) \leqslant \sup_{g\in\mathcal{G}} |\ell_{\mathcal{T}}(g) - \ell(g)| + \ell_{\mathcal{T}}(g^{\mathcal{G}}) - \ell(g^{\mathcal{G}}),$$

where we used the notation sup (supremum) for the least upper bound. Since $\mathbb{E}\ell_{\mathcal{T}}(g) = \mathbb{E}\ell(g)$, we obtain, after taking expectations on both sides of the inequality above:

$$\mathbb{E}\, \ell(g^{\mathcal{G}}_{\mathcal{T}}) - \ell(g^{\mathcal{G}}) \leqslant \mathbb{E} \sup_{g\in\mathcal{G}} |\ell_{\mathcal{T}}(g) - \ell(g)|.$$

HOEFFDING'S INEQUALITY

(b) If X is a zero-mean random variable taking values in the interval $[l, u]$, then the following *Hoeffding's inequality* states that the moment generating function satisfies

$$\mathbb{E}\, \mathrm{e}^{tX} \leqslant \exp\left(\frac{t^2(u-l)^2}{8}\right), \quad t \in \mathbb{R}. \tag{2.58}$$

Prove this result by using the fact that the line segment joining points $(l, \exp(tl))$ and $(u, \exp(tu))$ bounds the convex function $x \mapsto \exp(tx)$ for $x \in [l, u]$; that is:

$$\mathrm{e}^{tx} \leqslant \mathrm{e}^{tl}\frac{u-x}{u-l} + \mathrm{e}^{tu}\frac{x-l}{u-l}, \quad x \in [l, u].$$

(c) Let $Z_1, \ldots, Z_n$ be (possibly dependent and non-identically distributed) zero-mean random variables with moment generating functions that satisfy $\mathbb{E}\exp(tZ_k) \leqslant \exp(t^2\eta^2/2)$ for all k and some parameter η. Use Jensen's inequality (2.56) to prove that for any

☞ 429

$t > 0$,

$$\mathbb{E} \max_k Z_k = \frac{1}{t} \mathbb{E} \ln \max_k \mathrm{e}^{tZ_k} \leqslant \frac{1}{t} \ln n + \frac{t\eta^2}{2}.$$

From this derive that

$$\mathbb{E} \max_k Z_k \leqslant \eta \sqrt{2 \ln n}.$$

Finally, show that this last inequality implies that

$$\mathbb{E} \max_k |Z_k| \leqslant \eta \sqrt{2 \ln(2n)}. \tag{2.59}$$

(d) Returning to the objective of this exercise, denote the elements of $\mathcal{G}$ by $g_1, \ldots, g_{|\mathcal{G}|}$, and let $Z_k = \ell_{\mathcal{T}_n}(g_k) - \ell(g_k)$. By part (a) it is sufficient to bound $\mathbb{E} \max_k |Z_k|$. Show that the $\{Z_k\}$ satisfy the conditions of (c) with $\eta = (u - l)/\sqrt{n}$. For this you will need to apply part (b) to the random variable $\mathrm{Loss}(g(\boldsymbol{X}), Y) - \ell(g)$, where $(\boldsymbol{X}, Y)$ is a generic data point. Now complete the proof of (2.57).

17. Consider the problem in Exercise 16a above. Show that

$$|\ell_{\mathcal{T}}(g_{\mathcal{T}}^{\mathcal{G}}) - \ell(g^{\mathcal{G}})| \leqslant 2 \sup_{g \in \mathcal{G}} |\ell_{\mathcal{T}}(g) - \ell(g)| + \ell_{\mathcal{T}}(g^{\mathcal{G}}) - \ell(g^{\mathcal{G}}).$$

From this, conclude:

$$\mathbb{E} \, |\ell_{\mathcal{T}}(g_{\mathcal{T}}^{\mathcal{G}}) - \ell(g^{\mathcal{G}})| \leqslant 2 \mathbb{E} \sup_{g \in \mathcal{G}} |\ell_{\mathcal{T}}(g) - \ell(g)|.$$

The last bound allows us to assess how close the training loss $\ell_{\mathcal{T}}(g_{\mathcal{T}}^{\mathcal{G}})$ is to the optimal risk $\ell(g^{\mathcal{G}})$ within class $\mathcal{G}$.

18. Show that for the normal linear model $\boldsymbol{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$, the maximum likelihood estimator of σ^2 is identical to the method of moments estimator (2.37).

19. Let $X \sim \mathsf{Gamma}(\alpha, \lambda)$. Show that the pdf of $Z = 1/X$ is equal to

$$\frac{\lambda^\alpha (z)^{-\alpha-1} \mathrm{e}^{-\lambda (z)^{-1}}}{\Gamma(\alpha)}, \quad z > 0.$$

20. Consider the sequence $w_0, w_1, \ldots$, where $w_0 = g(\boldsymbol{\theta})$ is a non-degenerate initial guess and $w_t(\boldsymbol{\theta}) \propto w_{t-1}(\boldsymbol{\theta}) g(\tau \,|\, \boldsymbol{\theta})$, $t > 1$. We assume that $g(\tau \,|\, \boldsymbol{\theta})$ is not the constant function (with respect to $\boldsymbol{\theta}$) and that the maximum likelihood value

$$g(\tau \,|\, \widehat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta}} g(\tau \,|\, \boldsymbol{\theta}) < \infty$$

exists (is bounded). Let

$$l_t := \int g(\tau \,|\, \boldsymbol{\theta}) w_t(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}.$$

Show that $\{l_t\}$ is a strictly increasing and bounded sequence. Hence, conclude that its limit is $g(\tau \,|\, \widehat{\boldsymbol{\theta}})$.

21. Consider the Bayesian model for $\tau = \{x_1, \ldots, x_n\}$ with likelihood $g(\tau \,|\, \mu)$ such that $(X_1, \ldots, X_n \,|\, \mu) \sim_{\text{iid}} \mathcal{N}(\mu, 1)$ and prior pdf $g(\mu)$ such that $\mu \sim \mathcal{N}(\nu, 1)$ for some hyperparameter ν. Define a sequence of densities $w_t(\mu), t \geqslant 2$ via $w_t(\mu) \propto w_{t-1}(\mu)\, g(\tau \,|\, \mu)$, starting with $w_1(\mu) = g(\mu)$. Let a_t and b_t denote the mean and precision[4] of μ under the posterior $g_t(\mu \,|\, \tau) \propto g(\tau \,|\, \mu) w_t(\mu)$. Show that $g_t(\mu \,|\, \tau)$ is a normal density with precision $b_t = b_{t-1} + n,\ b_0 = 1$ and mean $a_t = (1 - \gamma_t) a_{t-1} + \gamma_t \overline{x}_n,\ a_0 = \nu$, where $\gamma_t := n/(b_{t-1} + n)$. Hence, deduce that $g_t(\mu \,|\, \tau)$ converges to a degenerate density with a point-mass at $\overline{x}_n$.

22. Consider again Example 2.8, where we have a normal model with improper prior $g(\boldsymbol{\theta}) = g(\mu, \sigma^2) \propto 1/\sigma^2$. Show that the prior predictive pdf is an improper density $g(x) \propto 1$, but that the posterior predictive density is

$$g(x \,|\, \tau) \propto \left(1 + \frac{(x - \overline{x}_n)^2}{(n+1) S_n^2}\right)^{-n/2}.$$

Deduce that $\frac{X - \overline{x}_n}{S_n \sqrt{(n+1)/(n-1)}} \sim \mathsf{t}_{n-1}$.

23. Assuming that $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{\text{iid}}{\sim} f$, show that (2.48) holds and that $\ell_n^* = -n\, \mathbb{E} \ln f(\boldsymbol{X})$.

24. Suppose that $\tau = \{x_1, \ldots, x_n\}$ are observations of iid continuous and strictly positive random variables, and that there are two possible models for their pdf. The first model $p = 1$ is

$$g(x \,|\, \theta, p = 1) = \theta \exp(-\theta x)$$

and the second $p = 2$ is

$$g(x \,|\, \theta, p = 2) = \left(\frac{2\theta}{\pi}\right)^{1/2} \exp\left(-\frac{\theta x^2}{2}\right).$$

For both models, assume that the prior for θ is a gamma density

$$g(\theta) = \frac{b^t}{\Gamma(t)} \theta^{t-1} \exp(-b\theta),$$

with the same hyperparameters b and t. Find a formula for the Bayes factor, $g(\tau \,|\, p = 1)/g(\tau \,|\, p = 2)$, for comparing these models.

25. Suppose that we have a total of m possible models with prior probabilities $g(p), p = 1, \ldots, m$. Show that the posterior probability of model $g(p \,|\, \tau)$ can be expressed in terms of all the $p(p-1)$ Bayes factors:

$$g(p = i \,|\, \tau) = \left(1 + \sum_{j \neq i} \frac{g(p = j)}{g(p = i)} B_{j|i}\right)^{-1}.$$

[4] The precision is the reciprocal of the variance.

26. Given the data $\tau = \{x_1, \ldots, x_n\}$, suppose that we use the likelihood $(X \,|\, \boldsymbol{\theta}) \sim \mathcal{N}(\mu, \sigma^2)$ with parameter $\boldsymbol{\theta} = (\mu, \sigma^2)^\top$ and wish to compare the following two nested models.

(a) Model $p = 1$, where $\sigma^2 = \sigma_0^2$ is known and this is incorporated via the prior

$$g(\boldsymbol{\theta} \,|\, p = 1) = g(\mu \,|\, \sigma^2, p = 1)\, g(\sigma^2 \,|\, p = 1) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(\mu - x_0)^2}{2\sigma^2}} \times \delta(\sigma^2 - \sigma_0^2).$$

(b) Model $p = 2$, where both mean and variance are unknown with prior

$$g(\boldsymbol{\theta} \,|\, p = 2) = g(\mu \,|\, \sigma^2)\, g(\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(\mu - x_0)^2}{2\sigma^2}} \times \frac{b^t (\sigma^2)^{-t-1} \mathrm{e}^{-b/\sigma^2}}{\Gamma(t)}.$$

Show that the prior $g(\boldsymbol{\theta} \,|\, p = 1)$ can be viewed as the limit of the prior $g(\boldsymbol{\theta} \,|\, p = 2)$ when $t \to \infty$ and $b = t\sigma_0^2$. Hence, conclude that

$$g(\tau \,|\, p = 1) = \lim_{\substack{t \to \infty \\ b = t\sigma_0^2}} g(\tau \,|\, p = 2)$$

and use this result to calculate $B_{1\,|\,2}$. Check that the formula for $B_{1\,|\,2}$ agrees with the Savage–Dickey density ratio:

$$\frac{g(\tau \,|\, p = 1)}{g(\tau \,|\, p = 2)} = \frac{g(\sigma^2 = \sigma_0^2 \,|\, \tau)}{g(\sigma^2 = \sigma_0^2)},$$

where $g(\sigma^2 \,|\, \tau)$ and $g(\sigma^2)$ are the posterior and prior, respectively, under model $p = 2$.