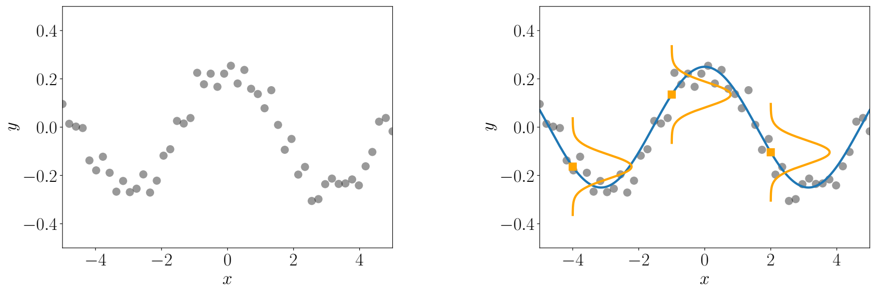
9

Linear Regression

In the following, we will apply the mathematical concepts from Chapters 2, 5, 6 and 7 to solving linear regression (curve fitting) problems. In *regression*, we want to find a function f that maps inputs $\boldsymbol{x} \in \mathbb{R}^D$ to corresponding function values $f(\boldsymbol{x}) \in \mathbb{R}$ given a set of training inputs $\boldsymbol{x}_n$ and corresponding observations $y_n = f(\boldsymbol{x}_n) + \epsilon$, where ϵ is a random variable that comprises measurement noise and unmodeled processes. An illustration of such a regression problem is given in Figure 9.1. A typical regression problem is given in Figure 9.1(a): For some input values x we observe (noisy) function values $y = f(x) + \epsilon$. The task is to infer the function f that generated the data. A possible solution is given in Figure 9.1(b), where we also show three distributions centered at the function values $f(x)$ that represent the noise in the data.

regression

Regression is a fundamental problem in machine learning, and regression problems appear in a diverse range of research areas and applications, including time-series analysis (e.g., system identification), control and robotics (e.g., reinforcement learning, forward/inverse model learning), optimization (e.g., line searches, global optimization), and deep-learning applications (e.g., computer games, speech-to-text translation, image recognition, automatic video annotation). Regression is also a key ingredient of classification algorithms.



(a) Regression problem: Observed noisy function values from which we wish to infer the underlying function that generated the data.

(b) Regression solution: Possible function that could have generated the data (blue) with indication of the measurement noise of the function value at the corresponding inputs (orange distributions).

Figure 9.1
(a) Dataset;
(b) Possible solution to the regression problem.

Finding a regression function requires solving a variety of problems, including

- **Choice of the model (type) and the parametrization** of the regression function. Given a data set, what function classes (e.g., polynomials) are good candidates for modeling the data, and what particular parametrization (e.g., degree of the polynomial) should we choose? Model selection, as discussed in Section 8.5, allows us to compare various models to find the simplest model that explains the training data reasonably well.
- **Finding good parameters.** Having chosen a model of the regression function, how do we find good model parameters? Here, we will need to look at different loss/objective functions (they determine what a "good" fit is) and optimization algorithms that allow us to minimize this loss.
- **Overfitting and model selection.** Overfitting is a problem when the regression function fits the training data "too well" but does not generalize to unseen test data. Overfitting typically occurs if the underlying model (or its parametrization) is overly flexible and expressive, see Section 8.5. We will look at the underlying reasons and discuss ways to mitigate the effect of overfitting in the context of linear regression.
- **Relationship between loss functions and parameter priors.** Loss functions (optimization objectives) are often motivated and induced by probabilistic models. We will look at the connection between loss functions and the underlying prior assumptions that induce these losses.
- **Uncertainty modeling.** In any practical setting, we have access to only a finite, potentially large, amount of (training) data for selecting the model class and the corresponding parameters. Given that this finite amount of training data does not cover all possible scenarios, we may want to describe the remaining parameter uncertainty to obtain a measure of confidence of the model's prediction at test time; the smaller the training set the more important uncertainty modeling. Consistent modeling of uncertainty equips model predictions with confidence bounds.

In the following, we will be using the mathematical tools from Chapters 3, 5, 6 and 7 to solve linear regression problems. We will discuss maximum likelihood and maximum a posteriori (MAP) estimation to find optimal model parameters. Using these parameter estimates, we will have a brief look at generalization errors and overfitting. Toward the end of this chapter, we will discuss Bayesian linear regression, which allows us to reason about model parameters at a higher level, thereby removing some of the problems encountered in maximum likelihood and MAP estimation.

9.1 Problem Formulation

We consider a regression problem with the likelihood function

$$p(y \,|\, \boldsymbol{x}) = \mathcal{N}\big(y \,|\, f(\boldsymbol{x}),\, \sigma^2\big)\,. \tag{9.1}$$

Here, $\boldsymbol{x} \in \mathbb{R}^D$ are inputs and $y \in \mathbb{R}$ are noisy function values (targets). With (9.1) the functional relationship between $\boldsymbol{x}$ and y is given as

$$y = f(\boldsymbol{x}) + \epsilon\,, \tag{9.2}$$

where $\epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)$ is independent, identically distributed (i.i.d.) measurement noise. In this particular case, ϵ is Gaussian distributed with mean 0 and variance σ^2. Our objective is to find a function that is close (similar) to the unknown function f that generated the data.

In this chapter, we focus on parametric models, i.e., we choose a parametrized function and find parameters $\boldsymbol{\theta}$ that "work well" for modeling the data. For the time being, we assume that the noise variance σ^2 is known and focus on learning the model parameters $\boldsymbol{\theta}$. In linear regression, we consider the special case that the parameters appear linearly in our model. An example of linear regression is given by

$$p(y \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\big(y \,|\, \boldsymbol{x}^\top\boldsymbol{\theta},\, \sigma^2\big) \tag{9.3}$$

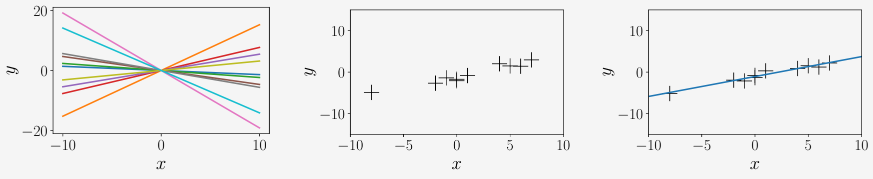
$$\iff y = \boldsymbol{x}^\top\boldsymbol{\theta} + \epsilon\,, \tag{9.4}$$

where $\boldsymbol{\theta} \in \mathbb{R}^D$ are the parameters we seek. The class of functions described by (9.4) are straight lines that pass through the origin. In (9.4), we chose a parametrization $f(\boldsymbol{x}) = \boldsymbol{x}^\top\boldsymbol{\theta}$.

likelihood

The *likelihood* in (9.3) is the probability density function of y evaluated at $\boldsymbol{x}^\top\boldsymbol{\theta}$. Note that the only source of uncertainty originates from the observation noise (as $\boldsymbol{x}$ and $\boldsymbol{\theta}$ are assumed known in (9.3)). Without observation noise the relationship between $\boldsymbol{x}$ and y would be deterministic and (9.3) would be a Delta distribution.

Example 9.1



(a) Example functions (straight lines) that can be described using the linear model in (9.4).

(b) Training set.

(c) Maximum likelihood estimate.

Figure 9.2 Linear regression example. (a) Example functions that fall into this category. (b) Training set. (c) Maximum likelihood estimate.

For $x, \theta \in \mathbb{R}$ the linear regression model in (9.4) describes straight lines (linear functions), and the parameter θ is the slope of the line. Figure 9.3(a) shows some example functions for different values of θ.

Linear regression refers to models that are linear in the parameters.

The linear regression model in (9.3)–(9.4) is not only linear in the parameters, but also linear in the inputs x. We will see later that $y = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta}$ for nonlinear transformations $\boldsymbol{\phi}$ is also a linear regression model because "linear regression" refers to models that are "linear in the parameters", i.e., models that describe a function by a linear combination of input features. Here, a "feature" is a representation $\boldsymbol{\phi}(\boldsymbol{x})$ of the inputs $\boldsymbol{x}$.

In the following, we will discuss in more detail how to find good parameters $\boldsymbol{\theta}$ and how to evaluate whether a parameter set "works well". For the time being we assume that the noise variance σ^2 is known.

9.2 Parameter Estimation

training set

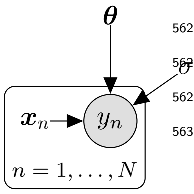
Consider the linear regression setting (9.4) and assume we are given a *training set* $\mathcal{D}$ consisting of N inputs $\boldsymbol{x}_n \in \mathbb{R}^D$ and corresponding observations/targets $y_n \in \mathbb{R}$, $n = 1, \dots, N$. The corresponding graphical model is given in Figure 9.3. Note that y_i and y_j are conditionally independent given their respective inputs $\boldsymbol{x}_i, \boldsymbol{x}_j$ so that the likelihood factorizes according to

$$p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = p(y_1, \dots, y_N \mid \boldsymbol{x}_1, \dots, \boldsymbol{x}_N, \boldsymbol{\theta}) \tag{9.5a}$$

$$= \prod_{n=1}^{N} p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}\big(y_n \mid \boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)\,, \tag{9.5b}$$

where we defined $\mathcal{X} := \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and $\mathcal{Y} := \{y_1, \dots, y_N\}$ as the sets of training inputs and corresponding targets, respectively. The likelihood and the factors $p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$ are Gaussian due to the noise distribution, see (9.3).

Figure 9.3 Probabilistic graphical model for linear regression. Observed random variables are shaded, deterministic/ known values are without circles.



In the following, we will discuss how to find optimal parameters $\boldsymbol{\theta}^* \in \mathbb{R}^D$ for the linear regression model (9.4). Once the parameters $\boldsymbol{\theta}^*$ are found, we can predict function values by using this parameter estimate in (9.4) so that at an arbitrary test input $\boldsymbol{x}_*$ we predict the probability of the corresponding target y_* as

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{\theta}^*) = \mathcal{N}\big(y_* \mid \boldsymbol{x}_*^\top \boldsymbol{\theta}^*,\, \sigma^2\big)\,. \tag{9.6}$$

In the following, we will have a look at parameter estimation by maximizing the likelihood, a topic that we already covered to some degree in Section 8.2.

9.2.1 Maximum Likelihood Estimation

maximum likelihood estimation

A widely used approach to finding the desired parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ is *maximum* *likelihood estimation* where we find parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ that maximize the likelihood (9.5b). Intuitively, maximizing the likelihood means maximizing the predictive distribution of the training data given the model parameters. We obtain the maximum likelihood parameters as

Maximizing the likelihood means maximizing the predictive distribution of the (training) data given the parameters.

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) \,. \tag{9.7}$$

Remark. The likelihood $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})$ is not a probability distribution in $\boldsymbol{\theta}$: It is simply a function of the parameters $\boldsymbol{\theta}$ but does not integrate to 1 (i.e., it is unnormalized), and may not even be integrable with respect to $\boldsymbol{\theta}$. However, the likelihood in (9.7) is a normalized probability distribution in $\boldsymbol{y}$. ♢

The likelihood is not a probability distribution in the parameters.

To find the desired parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ that maximize the likelihood, we typically perform gradient ascent (or gradient descent on the negative likelihood). In the case of linear regression we consider here, however, a closed-form solution exists, which makes iterative gradient descent unnecessary. In practice, instead of maximizing the likelihood directly, we apply the log-transformation to the likelihood function and minimize the negative log-likelihood.

Since the logarithm is a (strictly) monotonically increasing function, the optimum of a function f is identical to the optimum of $\log f$.

Remark (Log Transformation). Since the likelihood (9.5b) is a product of N Gaussian distributions, the log-transformation is useful since a) it does not suffer from numerical underflow, b) the differentiation rules will turn out simpler. More specifically, numerical underflow will be a problem when we multiply N probabilities, where N is the number of data points, since we cannot represent very small numbers, such as 10^{-256}. Furthermore, the log-transform will turn the product into a sum of log-probabilities such that the corresponding gradient is a sum of individual gradients, instead of a repeated application of the product rule (5.46) to compute the gradient of a product of N terms. ♢

To find the optimal parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ of our linear regression problem, we minimize the negative log-likelihood

$$-\log p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) = -\log \prod_{n=1}^{N} p(y_n \,|\, \boldsymbol{x}_n, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n \,|\, \boldsymbol{x}_n, \boldsymbol{\theta}) \,, \tag{9.8}$$

where we exploited that the likelihood (9.5b) factorizes over the number of data points due to our independence assumption on the training set.

In the linear regression model (9.4) the likelihood is Gaussian (due to the Gaussian additive noise term), such that we arrive at

$$\log p(y_n \,|\, \boldsymbol{x}_n, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const} \tag{9.9}$$

where the constant includes all terms independent of $\boldsymbol{\theta}$. Using (9.9) in the negative log-likelihood (9.8) we obtain (ignoring the constant terms)

$$\mathcal{L}(\boldsymbol{\theta}) := -\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 \tag{9.10a}$$

$$= \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right\|^2, \tag{9.10b}$$

The negative log-likelihood function is also called *error function*.

design matrix

where we define the *design matrix* $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$ as the collection of training inputs and $\boldsymbol{y} := [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$ as a vector that collects all training targets. Note that the nth row in the design matrix $\boldsymbol{X}$ corresponds to the training input $\boldsymbol{x}_n$.

In (9.10b) we used the fact that the sum of squared errors between the observations y_n and the corresponding model prediction $\boldsymbol{x}_n^\top \boldsymbol{\theta}$ equals the squared distance between $\boldsymbol{y}$ and $\boldsymbol{X}\boldsymbol{\theta}$. Remember from Section 3.1 that $\left\|\boldsymbol{x}\right\|^2 = \boldsymbol{x}^\top \boldsymbol{x}$ if we choose the dot product as the inner product.

With (9.10b) we have now a concrete form of the negative log-likelihood function we need to optimize. We immediately see that (9.10b) is quadratic in $\boldsymbol{\theta}$. This means that we can find a unique global solution $\boldsymbol{\theta}_{\mathrm{ML}}$ for minimizing the negative log-likelihood $\mathcal{L}$. We can find the global optimum by computing the gradient of $\mathcal{L}$, setting it to $\boldsymbol{0}$ and solving for $\boldsymbol{\theta}$.

Using the results from Chapter 5, we compute the gradient of $\mathcal{L}$ with respect to the parameters as

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})\right) \tag{9.11a}$$

$$= \frac{1}{2\sigma^2}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\theta}\right) \tag{9.11b}$$

$$= \frac{1}{\sigma^2}(-\boldsymbol{y}^\top\boldsymbol{X} + \boldsymbol{\theta}^\top\boldsymbol{X}^\top\boldsymbol{X}) \in \mathbb{R}^{1 \times D}. \tag{9.11c}$$

The maximum likelihood estimator $\boldsymbol{\theta}_{\mathrm{ML}}$ solves $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \boldsymbol{0}^\top$ (necessary optimality condition) and we obtain

Ignoring the possibility of duplicate data points, $\mathrm{rk}(\boldsymbol{X}) = D$ if $N \geqslant D$, i.e., we do not have more parameters than data points.

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \boldsymbol{0}^\top \overset{(9.11c)}{\iff} \boldsymbol{\theta}_{\mathrm{ML}}^\top\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{y}^\top\boldsymbol{X} \tag{9.12a}$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}}^\top = \boldsymbol{y}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} \tag{9.12b}$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}\,. \tag{9.12c}$$

We could right-multiply the first equation by $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}$ because $\boldsymbol{X}^\top\boldsymbol{X}$ is positive definite if $\mathrm{rk}(\boldsymbol{X}) = D$, where $\mathrm{rk}(\boldsymbol{X})$ denotes the rank of $\boldsymbol{X}$.

Remark. Setting the gradient to $\boldsymbol{0}^\top$ is a necessary and sufficient condition and we obtain a global minimum since the Hessian $\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{X}^\top\boldsymbol{X} \in \mathbb{R}^{D \times D}$ is positive definite. ♢

Example 9.2 (Fitting Lines)
Let us have a look at Figure 9.2, where we aim to fit a straight line $f(x) = \theta x$, where θ is an unknown slope, to a data set using maximum likelihood estimation. Examples of functions in this model class (straight lines) are shown in Figure 9.3(a). For the data set shown in Figure 9.3(b) we find the maximum likelihood estimate of the slope parameter θ using (9.12c) and obtain the maximum likelihood linear function in Figure 9.3(c).

5677 *Maximum Likelihood Estimation with Features*

So far, we considered the linear regression setting described in (9.4), which allowed us to fit straight lines to data using maximum likelihood estimation. However, straight lines are not sufficiently expressive when it comes to fitting more interesting data. Fortunately, linear regression offers us a way to fit nonlinear functions within the linear regression framework: Since "linear regression" only refers to "linear in the parameters", we can perform an arbitrary nonlinear transformation $\boldsymbol{\phi}(\boldsymbol{x})$ of the inputs $\boldsymbol{x}$ and then linearly combine the components of this transformation. The corresponding linear regression model is

Linear regression refers to "linear-in-the-parameters" regression models, but the inputs can undergo any nonlinear transformation.

$$\begin{aligned} p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) &= \mathcal{N}\big(y \mid \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta},\, \sigma^2\big) \\ \iff y &= \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta} + \epsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \epsilon\,, \end{aligned} \tag{9.13}$$

5678 where $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^K$ is a (nonlinear) transformation of the inputs $\boldsymbol{x}$ and 5679 $\phi_k : \mathbb{R}^D \to \mathbb{R}$ is the kth component of the *feature vector* $\boldsymbol{\phi}$. Note that the 5680 model parameters $\boldsymbol{\theta}$ still appear only linearly.

feature vector

Example 9.3 (Polynomial Regression)
We are concerned with a regression problem $y = \boldsymbol{\phi}^\top(x)\boldsymbol{\theta} + \epsilon$, where $x \in \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}^K$. A transformation that is often used in this context is

$$\boldsymbol{\phi}(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K-1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K\,. \tag{9.14}$$

This means, we "lift" the original one-dimensional input space into a K-dimensional feature space consisting of all monomials x^k for $k = 0, \ldots, K-1$. With these features, we can model polynomials of degree $\leqslant K-1$ within the framework of linear regression: A polynomial of degree $K-1$ is

$$f(x)=\sum_{k=0}^{K-1}\theta_k x^k=\boldsymbol{\phi}^\top(x)\boldsymbol{\theta} \tag{9.15}$$

where $\boldsymbol{\phi}$ is defined in (9.14) and $\boldsymbol{\theta}=[\theta_0,\dots,\theta_{K-1}]^\top\in\mathbb{R}^K$ contains the (linear) parameters θ_k.

feature matrix

design matrix

Let us now have a look at maximum likelihood estimation of the parameters $\boldsymbol{\theta}$ in the linear regression model (9.13). We consider training inputs $\boldsymbol{x}_n\in\mathbb{R}^D$ and targets $y_n\in\mathbb{R}$, $n=1,\dots,N$, and define the *feature matrix* (*design matrix*) as

$$\boldsymbol{\Phi}:=\begin{bmatrix}\boldsymbol{\phi}^\top(\boldsymbol{x}_1)\\ \vdots\\ \boldsymbol{\phi}^\top(\boldsymbol{x}_N)\end{bmatrix}=\begin{bmatrix}\phi_0(\boldsymbol{x}_1) & \cdots & \phi_{K-1}(\boldsymbol{x}_1)\\ \phi_0(\boldsymbol{x}_2) & \cdots & \phi_{K-1}(\boldsymbol{x}_2)\\ \vdots & & \vdots\\ \phi_0(\boldsymbol{x}_N) & \cdots & \phi_{K-1}(\boldsymbol{x}_N)\end{bmatrix}\in\mathbb{R}^{N\times K}, \tag{9.16}$$

where $\Phi_{ij}=\phi_j(\boldsymbol{x}_i)$ and $\phi_j:\mathbb{R}^D\to\mathbb{R}$.

Example 9.4 (Feature Matrix for Second-order Polynomials)
For a second-order polynomial and N training points $x_n\in\mathbb{R}, n=1,\dots,N$, the feature matrix is

$$\boldsymbol{\Phi}=\begin{bmatrix}1 & x_1 & x_1^2\\ 1 & x_2 & x_2^2\\ \vdots & \vdots & \vdots\\ 1 & x_N & x_N^2\end{bmatrix}. \tag{9.17}$$

With the feature matrix $\boldsymbol{\Phi}$ defined in (9.16) the negative log-likelihood for the linear regression model (9.13) can be written as

$$-\log p(\mathcal{Y}\,|\,\mathcal{X},\boldsymbol{\theta})=\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{\Phi\theta})^\top(\boldsymbol{y}-\boldsymbol{\Phi\theta})+\text{const}. \tag{9.18}$$

maximum likelihood estimate

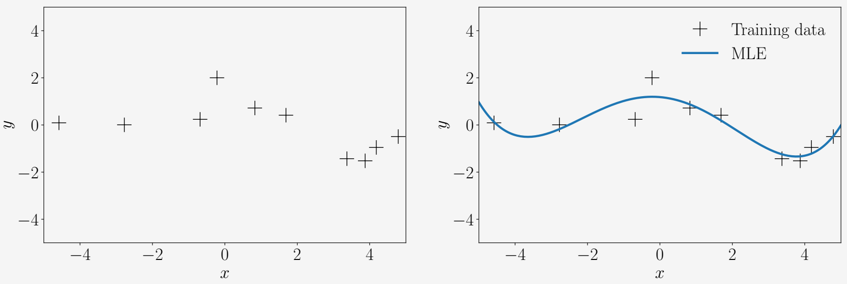
Comparing (9.18) with the negative log-likelihood in (9.10b) for the "feature-free" model, we immediately see we just need to replace $\boldsymbol{X}$ with $\boldsymbol{\Phi}$. Since both $\boldsymbol{X}$ and $\boldsymbol{\Phi}$ are independent of the parameters $\boldsymbol{\theta}$ that we wish to optimize, we arrive immediately at the *maximum likelihood estimate*

$$\boldsymbol{\theta}_{\text{ML}}=(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \tag{9.19}$$

for the linear regression problem with nonlinear features defined in (9.13).

Remark. When we were working without features, we required $\boldsymbol{X}^\top\boldsymbol{X}$ to be invertible, which is the case when the rows of $\boldsymbol{X}$ are linearly independent. In (9.19), we therefore require $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} \in \mathbb{R}^{D\times D}$ to be invertible. This is the case if and only if $\mathrm{rk}(\boldsymbol{\Phi}) = D$. $\diamondsuit$

Example 9.5 (Maximum Likelihood Polynomial Fit)



(a) Regression data set.

(b) Polynomial of degree 4 determined by maximum likelihood estimation.

Figure 9.4 Polynomial regression. (a) Data set consisting of (x_n, y_n) pairs, $n = 1, \ldots, 10$; (b) Maximum likelihood polynomial of degree 4.

Consider the data set in Figure 9.5(a). The data set consists of $N = 20$ pairs (x_n, y_n), where $x_n \sim \mathcal{U}[-5, 5]$ and $y_n = -\sin(x_n/5) + \cos(x_n) + \epsilon$, where $\epsilon \sim \mathcal{N}\big(0,\, 0.2^2\big)$.

We fit a polynomial of degree $K = 4$ using maximum likelihood estimation, i.e., parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ are given in (9.19). The maximum likelihood estimate yields function values $\boldsymbol{\phi}^\top(x_*)\boldsymbol{\theta}_{\mathrm{ML}}$ at any test location x_*. The result is shown in Figure 9.5(b).

Estimating the Noise Variance

Thus far, we assumed that the noise variance σ^2 is known. However, we can also use the principle of maximum likelihood estimation to obtain the maximum likelihood estimator σ^2_{ML} for the noise variance. To do this, we follow the standard procedure: we write down the log-likelihood, compute its derivative with respect to $\sigma^2 > 0$, set it to 0 and solve. The log-likelihood is given by

$$\log p(\mathcal{Y}\,|\,\mathcal{X},\boldsymbol{\theta},\sigma^2) = \sum_{n=1}^{N} \log \mathcal{N}\big(y_n\,|\,\boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta},\, \sigma^2\big) \tag{9.20a}$$

$$= \sum_{n=1}^{N}\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(y_n - \boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta})^2\right) \tag{9.20b}$$

$$= -\frac{N}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\underbrace{\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta})^2}_{=:s} + \text{const}\,. \tag{9.20c}$$

The partial derivative of the log-likelihood with respect to σ^2 is then

$$\frac{\partial \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} s = 0 \tag{9.21a}$$

$$\iff \frac{N}{2\sigma^2} = \frac{s}{2\sigma^4} \tag{9.21b}$$

so that we identify

$$\sigma_{\mathrm{ML}}^2 = \frac{s}{N} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta})^2 \,. \tag{9.22}$$

Therefore, the maximum likelihood estimate of the noise variance is the empirical mean of the squared distances between the noise-free function values $\boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta}$ and the corresponding noisy observations y_n at input locations $\boldsymbol{x}_n$.

9.2.2 Overfitting in Linear Regression

We just discussed how to use maximum likelihood estimation to fit linear models (e.g., polynomials) to data. We can evaluate the quality of the model by computing the error/loss incurred. One way of doing this is to compute the negative log-likelihood (9.10b), which we minimized to determine the maximum likelihood estimator. Alternatively, given that the noise parameter σ^2 is not a free model parameter, we can ignore the scaling by $1/\sigma^2$, so that we end up with a squared-error-loss function $\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2$. Instead of using this squared loss, we often use the *root mean squared error* (*RMSE*)

root mean squared error
RMSE

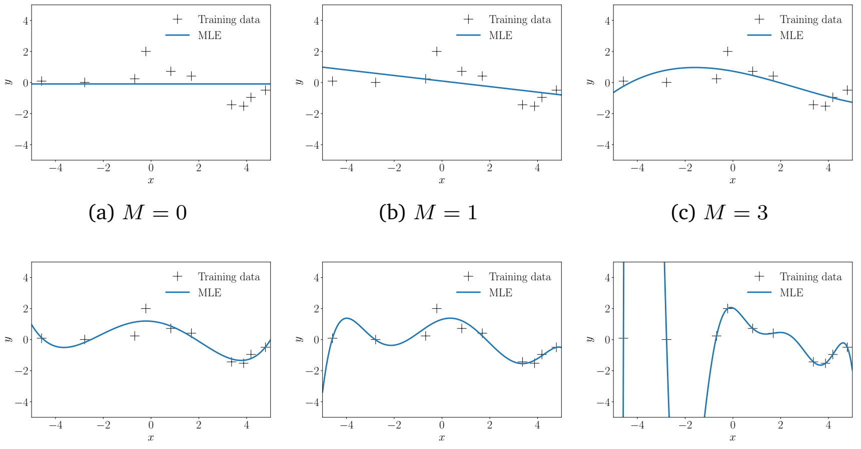
$$\sqrt{\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 / N} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}^\top(\boldsymbol{x}_n)\boldsymbol{\theta})^2} \,, \tag{9.23}$$

which (a) allows us to compare errors of data sets with different sizes and (b) has the same scale and the same units as the observed function values y_n. For example, if we fit a model that maps post-codes ($\boldsymbol{x}$ is given in latitude, longitude) to house prices (y-values are EUR) then the RMSE is also measured in EUR, whereas the squared error is given in EUR^2. If we choose to include the factor σ^2 from the original negative log-likelihood (9.10b) then we end up with a unitless objective, i.e., in the above example our objective would no longer be in EUR or EUR^2.

The RMSE is normalized.

The negative log-likelihood is unitless.

For model selection (see Section 8.5) we can use the RMSE (or the negative log-likelihood) to determine the best degree of the polynomial by finding the polynomial degree M that minimizes the objective. Given that the polynomial degree is a natural number, we can perform a brute-force search and enumerate all (reasonable) values of M. For a training set of size N it is sufficient to test $0 \leqslant M \leqslant N-1$. For $M \leqslant N$, the maximum likelihood estimator is unique. For $M > N$ we have more parameters



(a) $M = 0$ (b) $M = 1$ (c) $M = 3$

(d) $M = 4$ (e) $M = 6$ (f) $M = 9$

Figure 9.5 Maximum likelihood fits for different polynomial degrees M.

than data points, and would need to solve an underdetermined system of linear equations ($\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ in (9.19) would also no longer be invertible) so that there are infinitely many possible maximum likelihood estimators.

Figure 9.5 shows a number of polynomial fits determined by maximum likelihood for the dataset from Figure 9.5(a) with $N = 10$ observations. We notice that polynomials of low degree (e.g., constants ($M = 0$) or linear ($M = 1$) fit the data poorly and, hence, are poor representations of the true underlying function. For degrees $M = 3, \dots, 5$ the fits look plausible and smoothly interpolate the data. When we go to higher-degree polynomials, we notice that they fit the data better and better. In the extreme case of $M = N - 1 = 9$, the function will pass through every single data point. However, these high-degree polynomials oscillate wildly and are a poor representation of the underlying function that generated the data, such that we suffer from *overfitting*.

overfitting

Note that the noise variance $\sigma^2 > 0$.

Remember that the goal is to achieve good generalization by making accurate predictions for new (unseen) data. We obtain some quantitative insight into the dependence of the generalization performance on the polynomial of degree M by considering a separate test set comprising 200 data points generated using exactly the same procedure used to generate the training set. As test inputs, we chose a linear grid of 200 points in the interval of $[-5, 5]$. For each choice of M, we evaluate the RMSE (9.23) for both the training data and the test data.

Looking now at the test error, which is a qualitive measure of the generalization properties of the corresponding polynomial, we notice that initially the test error decreases, see Figure 9.6 (orange). For fourth-order polynomials the test error is relatively low and stays relatively constant up to degree 5. However, from degree 6 onward the test error increases significantly, and high-order polynomials have very bad generalization properties. In this particular example, this also is evident from the corresponding

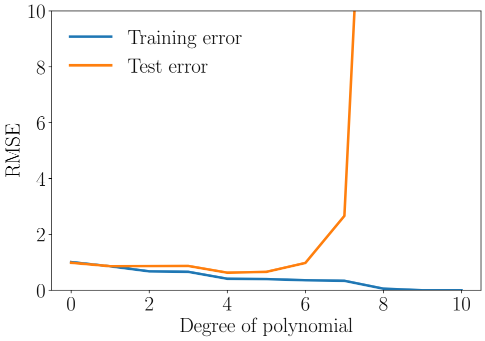


Figure 9.6 Training and test error.

training error

test error

maximum likelihood fits in Figure 9.5. Note that the *training error* (blue curve in Figure 9.6) never increases when the degree of the polynomial increases. In our example, the best generalization (the point of the smallest *test error*) is obtained for a polynomial of degree $M = 4$.

9.2.3 Maximum A Posteriori Estimation

We just saw that maximum likelihood estimation is prone to overfitting. We often observe that the magnitude of the parameter values becomes relatively large if we run into overfitting (Bishop, 2006).

To mitigate the effect of huge parameter values, we can place a prior distribution $p(\boldsymbol{\theta})$ on the parameters. The prior distribution explicitly encodes what parameter values are plausible (before having seen any data). For example, a Gaussian prior $p(\theta) = \mathcal{N}\big(0,\, 1\big)$ on a single parameter θ encodes that parameter values are expected lie in the interval $\mathbf{0} \pm 2$ (two standard deviations around the mean value). Once a dataset $\mathcal{X},\ \mathcal{Y}$ is available, instead of maximizing the likelihood we seek parameters that maximize the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y})$. This procedure is called *maximum a posteriori* (*MAP*) estimation.

maximum a posteriori

MAP

The posterior over the parameters $\boldsymbol{\theta}$, given the training data $\mathcal{X}, \mathcal{Y}$, is obtained by applying Bayes' theorem (Section 6.3) as

$$p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \,|\, \mathcal{X})}\,. \tag{9.24}$$

Since the posterior explicitly depends on the parameter prior $p(\boldsymbol{\theta})$, the prior will have an effect on parameter vector we find as the maximizer of the posterior. We will see this more explicitly in the following. The parameter vector $\boldsymbol{\theta}_{\text{MAP}}$ that maximizes the posterior (9.24) is the maximum a posteriori (MAP) estimate.

To find the MAP estimate, we follow steps that are similar in flavor to maximum likelihood estimation. We start with the log-transform and compute the log-posterior as

$$\log p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{const}\,, \tag{9.25}$$

where the constant comprises the terms that are independent of $\boldsymbol{\theta}$. We see that the log-posterior in (9.25) is the sum of the log-likelihood $p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta})$ and the log-prior $\log p(\boldsymbol{\theta})$ so that the MAP estimate will be a "compromise" between the prior (our suggestion for plausible parameter values before observing data) and the data-dependent likelihood.

To find the MAP estimate $\boldsymbol{\theta}_{\text{MAP}}$, we minimize the negative log-posterior distribution with respect to $\boldsymbol{\theta}$, i.e., we solve

$$\boldsymbol{\theta}_{\text{MAP}} \in \arg\min_{\boldsymbol{\theta}} \{-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\}\,. \tag{9.26}$$

The gradient of the negative log-posterior with respect to $\boldsymbol{\theta}$ is

$$-\frac{\mathrm{d}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})}{\mathrm{d}\boldsymbol{\theta}} = -\frac{\mathrm{d}\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}} - \frac{\mathrm{d}\log p(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\,, \tag{9.27}$$

where we identify the first term on the right-hand-side as the gradient of the negative log-likelihood from (9.11c).

With a (conjugate) Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0},\, b^2\boldsymbol{I})$ on the parameters $\boldsymbol{\theta}$, the negative log-posterior for the linear regression setting (9.13), we obtain the negative log posterior

$$-\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{const}\,. \tag{9.28}$$

Here, the first term corresponds to the contribution from the log-likelihood, and the second term originates from the log-prior. The gradient of the log-posterior with respect to the parameters $\boldsymbol{\theta}$ is then

$$-\frac{\mathrm{d}\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})}{\mathrm{d}\boldsymbol{\theta}} = \frac{1}{\sigma^2}(\boldsymbol{\theta}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{y}^\top\boldsymbol{\Phi}) + \frac{1}{b^2}\boldsymbol{\theta}^\top\,. \tag{9.29}$$

We will find the MAP estimate $\boldsymbol{\theta}_{\text{MAP}}$ by setting this gradient to $\mathbf{0}^\top$ and solving for $\boldsymbol{\theta}_{\text{MAP}}$. We obtain

$$\frac{1}{\sigma^2}(\boldsymbol{\theta}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{y}^\top\boldsymbol{\Phi}) + \frac{1}{b^2}\boldsymbol{\theta}^\top = \mathbf{0}^\top \tag{9.30a}$$

$$\iff \boldsymbol{\theta}^\top\left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{1}{b^2}\boldsymbol{I}\right) - \frac{1}{\sigma^2}\boldsymbol{y}^\top\boldsymbol{\Phi} = \mathbf{0}^\top \tag{9.30b}$$

$$\iff \boldsymbol{\theta}^\top\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right) = \boldsymbol{y}^\top\boldsymbol{\Phi} \tag{9.30c}$$

$$\iff \boldsymbol{\theta}^\top = \boldsymbol{y}^\top\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)^{-1} \tag{9.30d}$$

$\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ is symmetric, positive semidefinite and the additional term is strictly positive definite so that all eigenvalues of the matrix to be inverted are positive.

so that the MAP estimate is (by transposing both sides of the last equality)

$$\boldsymbol{\theta}_{\text{MAP}} = \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}\,. \tag{9.31}$$

Comparing the MAP estimate in (9.31) with the maximum likelihood estimate in (9.19) we see that the only difference between both solutions is the additional term $\frac{\sigma^2}{b^2}\boldsymbol{I}$ in the inverse matrix. This term ensures that

$\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}$ is symmetric and strictly positive definite (i.e., its inverse exists and the MAP estimate is the unique solution of a system of linear equations) and plays the role of the *regularizer*.

regularizer

Example 9.6 (MAP Estimation for Polynomial Regression)

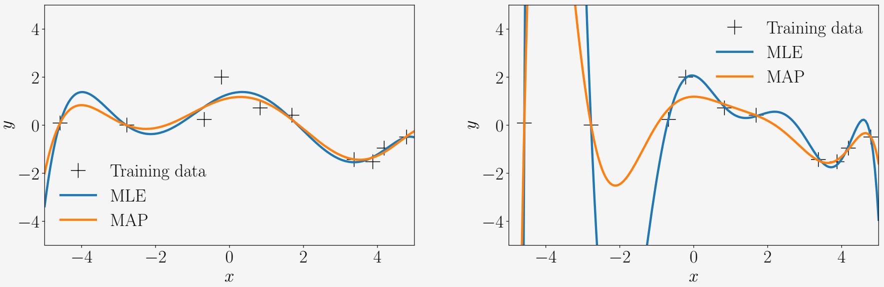


Figure 9.7 Polynomial regression: Maximum likelihood and MAP estimates.

In the polynomial regression example from Section 9.2.1, we place a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ on the parameters $\boldsymbol{\theta}$ and determine the MAP estimates according to (9.31). In Figure 9.7, we show both the maximum likelihood and the MAP estimates for polynomials of degree 6 (left) and degree 8 (right). The prior (regularizer) does not play a significant role for the low-degree polynomial, but keeps the function relatively smooth for higher-degree polynomials. Although the MAP estimate can push the boundaries of overfitting it is not a general solution to this problem so that we need a more principled approach to tackle overfitting.

9.2.4 MAP Estimation as Regularization

Instead of placing a prior distribution on the parameters $\boldsymbol{\theta}$ it is also possible to mitigate the effect of overfitting by penalizing the amplitude of the parameter by means of *regularization*. In *regularized least squares*, we consider the loss function

regularization

regularized least squares

$$\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \,, \tag{9.32}$$

which we minimize with respect to $\boldsymbol{\theta}$ (see Section 8.1.3). Here, the first term is a *data-fit term* (also called *misfit term*), which is proportional to the negative log-likelihood, see (9.10b). The second term is called the *regularizer*, and the *regularization parameter* $\lambda \geqslant 0$ controls the "strictness" of the regularization.

data-fit term
misfit term

regularizer

regularization parameter

Remark. Instead of the Euclidean norm $\|\cdot\|_2$, we can choose any p-norm $\|\cdot\|_p$ in (9.32). In practice, smaller values for p lead to sparser solutions. Here, "sparse" means that many parameter values $\theta_d = 0$, which is also useful for variable selection. For $p = 1$, the regularizer is called *LASSO*

LASSO

(least absolute shrinkage and selection operator) and was proposed by Tibshirani (1996). ♢

The regularizer $\lambda \|\boldsymbol{\theta}\|_2^2$ in (9.32) can be interpreted as a negative log-Gaussian prior, which we use in MAP estimation, see (9.26). More specifically, with a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{0},\, b^2\boldsymbol{I})$, we obtain the negative log-Gaussian prior

$$-\log p(\boldsymbol{\theta}) = \frac{1}{2b^2}\|\boldsymbol{\theta}\|_2^2 + \text{const} \tag{9.33}$$

so that for $\lambda = \frac{1}{2b^2}$ the regularization term and the negative log-Gaussian prior are identical.

Given that the regularized least-squares loss function in (9.32) consists of terms that are closely related to the negative log-likelihood plus a negative log-prior, it is not surprising that, when we minimize this loss, we obtain a solution that closely resembles the MAP estimate in (9.31). More specifically, minimizing the regularized least-squares loss function yields

$$\boldsymbol{\theta}_{\text{RLS}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}\,, \tag{9.34}$$

which is identical to the MAP estimate in (9.31) for $\lambda = \frac{\sigma^2}{b^2}$, where σ^2 is the noise variance and b^2 the variance of the (isotropic) Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{0},\, b^2\boldsymbol{I})$.

So far, we covered parameter estimation using maximum likelihood and MAP estimation where we found point estimates $\boldsymbol{\theta}^*$ that optimize an objective function (likelihood or posterior). We saw that both maximum likelihood and MAP estimation can lead to overfitting. In the next section, we will discuss Bayesian linear regression, where we use Bayesian inference (Section 8.3) to find a posterior distribution over the unknown parameters, which we subsequently use to make predictions. More specifically, for predictions we will average over all plausible sets of parameters instead of focusing on a point estimate.

9.3 Bayesian Linear Regression

Previously, we looked at linear regression models where we estimated the model parameters $\boldsymbol{\theta}$, e.g., by means of maximum likelihood or MAP estimation. We discovered that MLE can lead to severe overfitting, in particular, in the small-data regime. MAP addresses this issue by placing a prior on the parameters that plays the role of a regularizer.

Bayesian linear regression

Bayesian linear regression pushes the idea of the parameter prior a step further and does not even attempt to compute a point estimate of the parameters, but instead the full posterior over the parameters is taken into account when making predictions. This means we do not fit any parameters, but we compute an average over all plausible parameters settings (according to the posterior).

9.3.1 Model

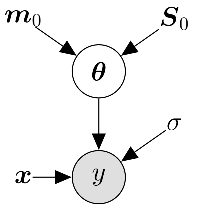
In Bayesian linear regression, we consider the model

$$\begin{aligned} &\text{prior} && p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)\,, \\ &\text{likelihood} && p(y \,|\, \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\big(y \,|\, \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta},\, \sigma^2\big)\,, \end{aligned} \tag{9.35}$$

where we now explicitly place a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$ on $\boldsymbol{\theta}$, which turns the parameter vector into a random variable. This allows us to write down the corresponding graphical model in Figure 9.8, where we made the parameters of the Gaussian prior on $\boldsymbol{\theta}$ explicit. The full probabilistic model, i.e., the joint distribution of observed and unobserved random variables, y and $\boldsymbol{\theta}$, respectively, is

$$p(y, \boldsymbol{\theta} \,|\, \boldsymbol{x}) = p(y \,|\, \boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})\,. \tag{9.36}$$

Figure 9.8 Graphical model for Bayesian linear regression.



9.3.2 Prior Predictions

In practice, we are usually not so much interested in the parameter values $\boldsymbol{\theta}$ themselves. Instead, our focus often lies in the predictions we make with those parameter values. In a Bayesian setting, we take the parameter distribution and average over all plausible parameter settings when we make predictions. More specifically, to make predictions at an input $\boldsymbol{x}_*$, we integrate out $\boldsymbol{\theta}$ and obtain

$$p(y_* \,|\, \boldsymbol{x}_*) = \int p(y_* \,|\, \boldsymbol{x}_*, \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}}[p(y_* \,|\, \boldsymbol{x}_*, \boldsymbol{\theta})]\,, \tag{9.37}$$

which we can interpret as the average prediction of $y_* \,|\, \boldsymbol{x}_*, \boldsymbol{\theta}$ for all plausible parameters $\boldsymbol{\theta}$ according to the prior distribution $p(\boldsymbol{\theta})$. Note that predictions using the prior distribution only require to specify the input $\boldsymbol{x}_*$, but no training data.

In our model (9.35), we chose a conjugate (Gaussian) prior on $\boldsymbol{\theta}$ so that the predictive distribution is Gaussian as well (and can be computed in closed form): With the prior distribution $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big)$, we obtain the predictive distribution as

$$p(y_* \,|\, \boldsymbol{x}_*) = \mathcal{N}\big(\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_0,\, \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_0\boldsymbol{\phi}(\boldsymbol{x}_*) + \sigma^2\big)\,, \tag{9.38}$$

where we exploited that (i) the prediction is Gaussian due to conjugacy (see Section 6.6) and the marginalization property of Gaussians (see Section 6.5), (ii) the Gaussian noise is independent so that

$$\mathbb{V}[y_*] = \mathbb{V}_{\boldsymbol{\theta}}[\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}] + \mathbb{V}_\epsilon[\epsilon]\,, \tag{9.39}$$

(iii) y_* is a linear transformation of $\boldsymbol{\theta}$ so that we can apply the rules for computing the mean and covariance of the prediction analytically by using (6.51) and (6.52), respectively. In (9.38), the term $\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_0\boldsymbol{\phi}(\boldsymbol{x}_*)$ in the predictive variance explicitly accounts for the uncertainty associated with the parameters $\boldsymbol{\theta}$, whereas σ^2 is the uncertainty contribution due to the measurement noise.

If we are interested in predicting noise-free function values $f(\boldsymbol{x}_*) = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}$ instead of the noise-corrupted targets y_* we obtain

$$p(f(\boldsymbol{x}_*)) = \mathcal{N}\big(\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_0,\ \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_0\boldsymbol{\phi}(\boldsymbol{x}_*)\big)\,, \tag{9.40}$$

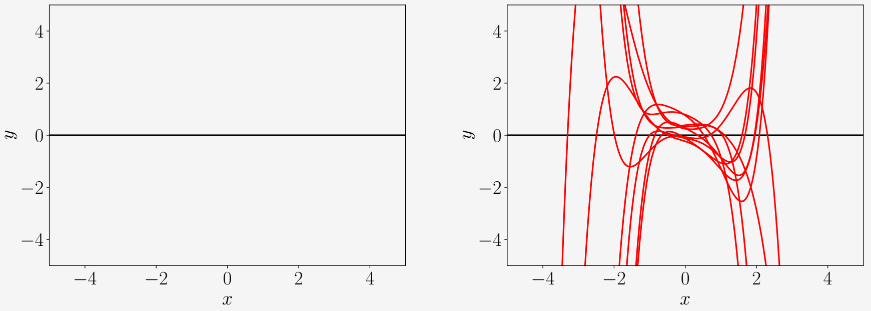
which only differs from (9.38) in the omission of the noise variance σ^2 in the predictive variance.

Remark (Distribution over Functions). Since we can represent the distribution $p(\boldsymbol{\theta})$ using a set of samples $\boldsymbol{\theta}_i$ and every sample $\boldsymbol{\theta}_i$ gives rise to a function $f_i(\cdot) = \boldsymbol{\theta}_i^\top\boldsymbol{\phi}(\cdot)$ it follows that the parameter distribution $p(\boldsymbol{\theta})$ induces a distribution $p(f(\cdot))$ over functions. Here we use the notation $(\cdot)$ to explicitly denote a functional relationship. ♢

The parameter distribution $p(\boldsymbol{\theta})$ induces a distribution over functions.

Example 9.7 (Prior over Functions)

Let us consider a Bayesian linear regression problem with polynomials of degree 5. We choose a parameter prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{0},\ \frac{1}{4}\boldsymbol{I}\big)$. Figure 9.9 visualizes the induced prior distribution over functions (shaded area: dark-gray: 67% confidence bound, light-gray: 95% confidence bound) induced by this parameter prior, including some function samples from this prior.



(a) Prior distribution over functions.

(b) Samples from the prior distribution over functions.

Figure 9.9 Prior over functions. (a) Distribution over functions represented by the mean function (black line) and the marginal uncertainties (shaded), representing the 67% and 95% confidence bounds, respectively; (b) Samples from the prior over functions, which are induced by the samples from the parameter prior.

A function sample is obtained by first sampling a parameter vector $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$ and then computing $f_i(\cdot) = \boldsymbol{\theta}_i^\top\boldsymbol{\phi}(\cdot)$. We used 200 input locations $x_* \in [-5, 5]$ to which we apply the feature function $\boldsymbol{\phi}(\cdot)$. The uncertainty (represented by the shaded area) in Figure 9.9 is solely due to the parameter uncertainty because we considered the noise-free predictive distribution (9.40).

So far, we looked at computing predictions using the parameter prior $p(\boldsymbol{\theta})$. However, when we have a parameter posterior (given some training data $\boldsymbol{X}, \boldsymbol{y}$), the same principles for prediction and inference hold as in (9.37) – we just need to replace the prior $p(\boldsymbol{\theta})$ with the posterior

$p(\boldsymbol{\theta} \,|\, \boldsymbol{X}, \boldsymbol{y})$. In the following, we will derive the posterior distribution in detail before using it to make predictions.

9.3.3 Posterior Distribution

Given a training set of inputs $\boldsymbol{x}_n \in \mathbb{R}^D$ and corresponding observations $y_n \in \mathbb{R}$, $n = 1, \ldots, N$, we compute the posterior over the parameters using Bayes' theorem as

$$p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \,|\, \mathcal{X})} \,, \tag{9.41}$$

where $\mathcal{X}$ is the set of training inputs and $\mathcal{Y}$ the collection of corresponding training targets. Furthermore, $p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta})$ is the likelihood, $p(\boldsymbol{\theta})$ the parameter prior and

$$p(\mathcal{Y} \,|\, \mathcal{X}) = \int p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}}[p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) \tag{9.42}$$

marginal likelihood
evidence
The marginal likelihood is the expected likelihood under the parameter prior.

the *marginal likelihood/evidence*, which is independent of the parameters $\boldsymbol{\theta}$ and ensures that the posterior is normalized, i.e., it integrates to 1. We can think of the marginal likelihood as the likelihood averaged over all possible parameter settings (with respect to the prior distribution $p(\boldsymbol{\theta})$).

Theorem 9.1 (Parameter Posterior)**.** *In our model* (9.35)*, the parameter posterior* (9.41) *can be computed in closed form as*

$$p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \mathcal{N}\big(\boldsymbol{\theta} \,|\, \boldsymbol{m}_N, \, \boldsymbol{S}_N\big) \,, \tag{9.43a}$$

$$\boldsymbol{S}_N = (\boldsymbol{S}_0^{-1} + \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \,, \tag{9.43b}$$

$$\boldsymbol{m}_N = \boldsymbol{S}_N (\boldsymbol{S}_0^{-1} \boldsymbol{m}_0 + \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{y}) \,, \tag{9.43c}$$

where the subscript N indicates the size of the training set.

Proof Bayes' theorem tells us that the posterior $p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y})$ is proportional to the product of the likelihood $p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$:

$$\text{posterior} \qquad p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \,|\, \mathcal{X})} \tag{9.44a}$$

$$\text{likelihood} \qquad p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{\Phi}\boldsymbol{\theta}, \, \sigma^2 \boldsymbol{I}\big) \tag{9.44b}$$

$$\text{prior} \qquad p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{\theta} \,|\, \boldsymbol{m}_0, \, \boldsymbol{S}_0\big) \,. \tag{9.44c}$$

Instead of looking at the product of the prior and the likelihood, we can transform the problem into log-space and solve for the mean and covariance of the posterior by completing the squares.

The sum of the log-prior and the log-likelihood is

$$\log \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{\Phi}\boldsymbol{\theta}, \, \sigma^2 \boldsymbol{I}\big) + \log \mathcal{N}\big(\boldsymbol{\theta} \,|\, \boldsymbol{m}_0, \, \boldsymbol{S}_0\big) \tag{9.45a}$$

$$= -\frac{1}{2}\big(\sigma^{-2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + (\boldsymbol{\theta} - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1} (\boldsymbol{\theta} - \boldsymbol{m}_0)\big) + \text{const} \tag{9.45b}$$

where the constant contains terms independent of $\boldsymbol{\theta}$. We will ignore the constant in the following. We now factorize (9.45b), which yields

$$
\begin{aligned}
&-\frac{1}{2}\big(\sigma^{-2}\boldsymbol{y}^\top\boldsymbol{y} - 2\sigma^{-2}\boldsymbol{y}^\top\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\boldsymbol{S}_0^{-1}\boldsymbol{\theta} \\
&\quad - 2\boldsymbol{m}_0^\top\boldsymbol{S}_0^{-1}\boldsymbol{\theta} + \boldsymbol{m}_0^\top\boldsymbol{S}_0^{-1}\boldsymbol{m}_0\big) \qquad (9.46a)\\
&= -\frac{1}{2}\big(\boldsymbol{\theta}^\top(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{S}_0^{-1})\boldsymbol{\theta} - 2(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{S}_0^{-1}\boldsymbol{m}_0)^\top\boldsymbol{\theta}\big) + \text{const}\,, \qquad (9.46b)
\end{aligned}
$$

where the constant contains the black terms in (9.46a), which are independent of $\boldsymbol{\theta}$. The orange terms are terms that are linear in $\boldsymbol{\theta}$, and the blue terms are the ones that are quadratic in $\boldsymbol{\theta}$. Inspecting (9.46b), we find that this equation is quadratic in $\boldsymbol{\theta}$. The fact that the unnormalized log-posterior distribution is a (negative) quadratic form implies that the posterior is Gaussian, i.e.,

$$
\begin{aligned}
p(\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y}) &= \exp(\log p(\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y})) \propto \exp(\log p(\mathcal{Y}\,|\,\mathcal{X},\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})) \qquad (9.47a)\\
&\propto \exp\Big(-\frac{1}{2}\big(\boldsymbol{\theta}^\top(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{S}_0^{-1})\boldsymbol{\theta} - 2(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{S}_0^{-1}\boldsymbol{m}_0)^\top\boldsymbol{\theta}\big)\Big)\,, \qquad (9.47b)
\end{aligned}
$$

5844 where we used (9.46b) in the last expression.

The remaining task is it to bring this (unnormalized) Gaussian into the form that is proportional to $\mathcal{N}(\boldsymbol{\theta}\,|\,\boldsymbol{m}_N,\,\boldsymbol{S}_N)$, i.e., we need to identify the mean $\boldsymbol{m}_N$ and the covariance matrix $\boldsymbol{S}_N$. To do this, we use the concept of *completing the squares*. The desired log-posterior is

completing the squares

$$
\begin{aligned}
\log\mathcal{N}(\boldsymbol{\theta}\,|\,\boldsymbol{m}_N,\,\boldsymbol{S}_N) &= -\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{m}_N)^\top\boldsymbol{S}_N^{-1}(\boldsymbol{\theta}-\boldsymbol{m}_N) + \text{const} \qquad (9.48a)\\
&= -\frac{1}{2}\big(\boldsymbol{\theta}^\top\boldsymbol{S}_N^{-1}\boldsymbol{\theta} - 2\boldsymbol{m}_N^\top\boldsymbol{S}_N^{-1}\boldsymbol{\theta} + \boldsymbol{m}_N^\top\boldsymbol{S}_N^{-1}\boldsymbol{m}_N\big)\,. \qquad (9.48b)
\end{aligned}
$$

Here, we factorized the quadratic form $(\boldsymbol{\theta}-\boldsymbol{m}_N)^\top\boldsymbol{S}_N^{-1}(\boldsymbol{\theta}-\boldsymbol{m}_N)$ into a term that is quadratic in $\boldsymbol{\theta}$ alone (blue), a term that is linear in $\boldsymbol{\theta}$ (orange), and a constant term (black). This allows us now to find $\boldsymbol{S}_N$ and $\boldsymbol{m}_N$ by matching the colored expressions in (9.46b) and (9.48b), which yields

$$
\boldsymbol{S}_N^{-1} = \boldsymbol{\Phi}^\top\sigma^{-2}\boldsymbol{I}\boldsymbol{\Phi} + \boldsymbol{S}_0^{-1} \iff \boldsymbol{S}_N = (\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{S}_0^{-1})^{-1}\,, \qquad (9.49)
$$

$$
\boldsymbol{m}_N^\top\boldsymbol{S}_N^{-1} = (\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{S}_0^{-1}\boldsymbol{m}_0)^\top \iff \boldsymbol{m}_N = \boldsymbol{S}_N(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{S}_0^{-1}\boldsymbol{m}_0)\,. \qquad (9.50)
$$

5845 □

Remark (General Approach to Completing the Squares). If we are given an equation

$$
\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{a}^\top\boldsymbol{x} + \text{const}_1\,, \qquad (9.51)
$$

where $\boldsymbol{A}$ is symmetric and positive definite, which we wish to bring into the form

$$(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}(\boldsymbol{x}-\boldsymbol{\mu}) + \text{const}_2\,, \tag{9.52}$$

we can do this by setting

$$\boldsymbol{\Sigma} := \boldsymbol{A}\,, \tag{9.53}$$

$$\boldsymbol{\mu} := \boldsymbol{\Sigma}^{-1}\boldsymbol{a} \tag{9.54}$$

and $\text{const}_2 = \text{const}_1 - \boldsymbol{\mu}^\top\boldsymbol{\Sigma}\boldsymbol{\mu}$. ♢

We can see that the terms inside the exponential in (9.47b) are of the form (9.51) with

$$\boldsymbol{A} := \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \boldsymbol{S}_0^{-1}\,, \tag{9.55}$$

$$\boldsymbol{a} := \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{y} + \boldsymbol{S}_0^{-1}\boldsymbol{m}_0\,. \tag{9.56}$$

Since $\boldsymbol{A}, \boldsymbol{a}$ can be difficult to identify in equations like (9.46a), it is often helpful to bring these equations into the form (9.51) that decouples quadratic term, linear terms and constants, which simplifies finding the desired solution.

9.3.4 Posterior Predictions

In (9.37), we computed the predictive distribution of y_* at a test input $\boldsymbol{x}_*$ using the parameter prior $p(\boldsymbol{\theta})$. In principle, predicting with the parameter posterior $p(\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y})$ is not fundamentally different given that in our conjugate model the prior and posterior are both Gaussian (with different parameters). Therefore, by following the same reasoning as in Section 9.3.2 we obtain the (posterior) predictive distribution

$$\begin{aligned} p(y_*\,|\,\mathcal{X},\mathcal{Y},\boldsymbol{x}_*) &= \int p(y_*\,|\,\boldsymbol{x}_*,\boldsymbol{\theta})p(\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y})\mathrm{d}\boldsymbol{\theta} && (9.57\text{a})\\ &= \int \mathcal{N}\big(y_*\,|\,\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta},\,\sigma^2\big)\mathcal{N}\big(\boldsymbol{\theta}\,|\,\boldsymbol{m}_N,\,\boldsymbol{S}_N\big)\mathrm{d}\boldsymbol{\theta} && (9.57\text{b})\\ &= \mathcal{N}\big(y_*\,|\,\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N,\,\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*)+\sigma^2\big)\,. && (9.57\text{c}) \end{aligned}$$

The term $\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*)$ reflects the posterior uncertainty associated with the parameters $\boldsymbol{\theta}$. Note that $\boldsymbol{S}_N$ depends on the training inputs through $\boldsymbol{\Phi}$, see (9.43b). The predictive mean $\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N$ coincides with the MAP estimate.

Remark (Marginal Likelihood and Posterior Predictive Distribution). By replacing the integral in (9.57a) the predictive distribution can be equivalently written as the expectation $\mathbb{E}_{\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y}}[p(y_*\,|\,\boldsymbol{x}_*,\boldsymbol{\theta})]$, where the expectation is taken with respect to the parameter posterior $p(\boldsymbol{\theta}\,|\,\mathcal{X},\mathcal{Y})$.

Writing the posterior predictive distribution in this way highlights a close resemblance to the marginal likelihood (9.42). The key difference between the marginal likelihood and the posterior predictive distribution are (i) the marginal likelihood can be thought of predicting the training targets $\boldsymbol{y}$ and not the test targets y_*, (ii) the marginal likelihood averages with respect to the parameter prior and not the parameter posterior. $\diamondsuit$

Remark (Mean and Variance of Noise-Free Function Values). In many cases, we are not interested in the predictive distribution $p(y_* \mid \mathcal{X}, \mathcal{Y}, \boldsymbol{x}_*)$ of a (noisy) observation y_*. Instead, we would like to obtain the distribution of the (noise-free) function values $f(\boldsymbol{x}_*) = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta}$. We determine the corresponding moments by exploiting the properties of means and variances, which yields

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x}_*) \mid \mathcal{X}, \mathcal{Y}] &= \mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}] = \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}] \\ &= \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N = \boldsymbol{m}_N^\top\boldsymbol{\phi}(\boldsymbol{x}_*)\,,\end{aligned} \tag{9.58}$$

$$\begin{aligned}\mathbb{V}_{\boldsymbol{\theta}}[f(\boldsymbol{x}_*) \mid \mathcal{X}, \mathcal{Y}] &= \mathbb{V}_{\boldsymbol{\theta}}[\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}] \\ &= \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\mathbb{V}_{\boldsymbol{\theta}}[\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}]\boldsymbol{\phi}(\boldsymbol{x}_*) \\ &= \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*)\end{aligned} \tag{9.59}$$

We see that the predictive mean is the same as the predictive mean for noisy observations as the noise has mean 0, and the predictive variance only differs by σ^2, which is the variance of the measurement noise: When we predict noisy function values, we need to include σ^2 as a source of uncertainty, but this term is not needed for noise-free predictions. Here, the only remaining uncertainty stems from the parameter posterior. $\diamondsuit$

Integrating out parameters induces a distribution over functions.

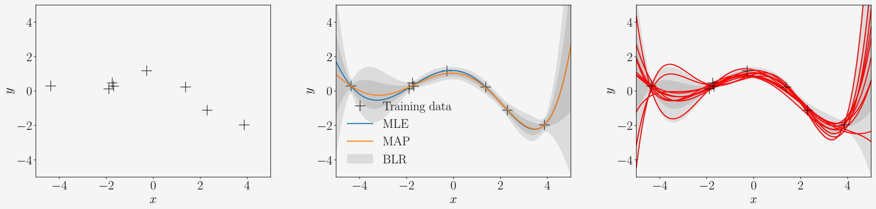
Remark (Distribution over Functions). The fact that we integrate out the parameters $\boldsymbol{\theta}$ induces a distribution over functions: If we sample $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})$ from the parameter posterior, we obtain a single function realization $\boldsymbol{\theta}_i^\top\boldsymbol{\phi}(\cdot)$. The *mean function*, i.e., the set of all expected function values $\mathbb{E}_{\boldsymbol{\theta}}[f(\cdot) \mid \boldsymbol{\theta}, \mathcal{X}, \mathcal{Y}]$, of this distribution over functions is $\boldsymbol{m}_N^\top\boldsymbol{\phi}(\cdot)$. The (marginal) variance, i.e., the variance of the function $f(\cdot)$, is given by $\boldsymbol{\phi}^\top(\cdot)\boldsymbol{S}_N\boldsymbol{\phi}(\cdot)$. $\diamondsuit$

mean function

Example 9.8 (Posterior over Functions)

Let us revisit the Bayesian linear regression problem with polynomials of degree 5. We choose a parameter prior $p(\boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{0},\, \frac{1}{4}\boldsymbol{I}\big)$. Figure 9.9 visualizes the prior over functions induced by the parameter prior and sample functions from this prior.

Figure 9.10 shows the posterior over functions that we obtain via Bayesian linear regression. The training dataset is shown in panel (a); Panel (b) shows the posterior distribution over functions, including the functions we would obtain via maximum likelihood and MAP estimation. The function we obtain using the MAP estimate also corresponds to the posterior mean function in the Bayesian linear regression setting. Panel (c) shows some plausible realizations (samples) of functions under that posterior over functions.



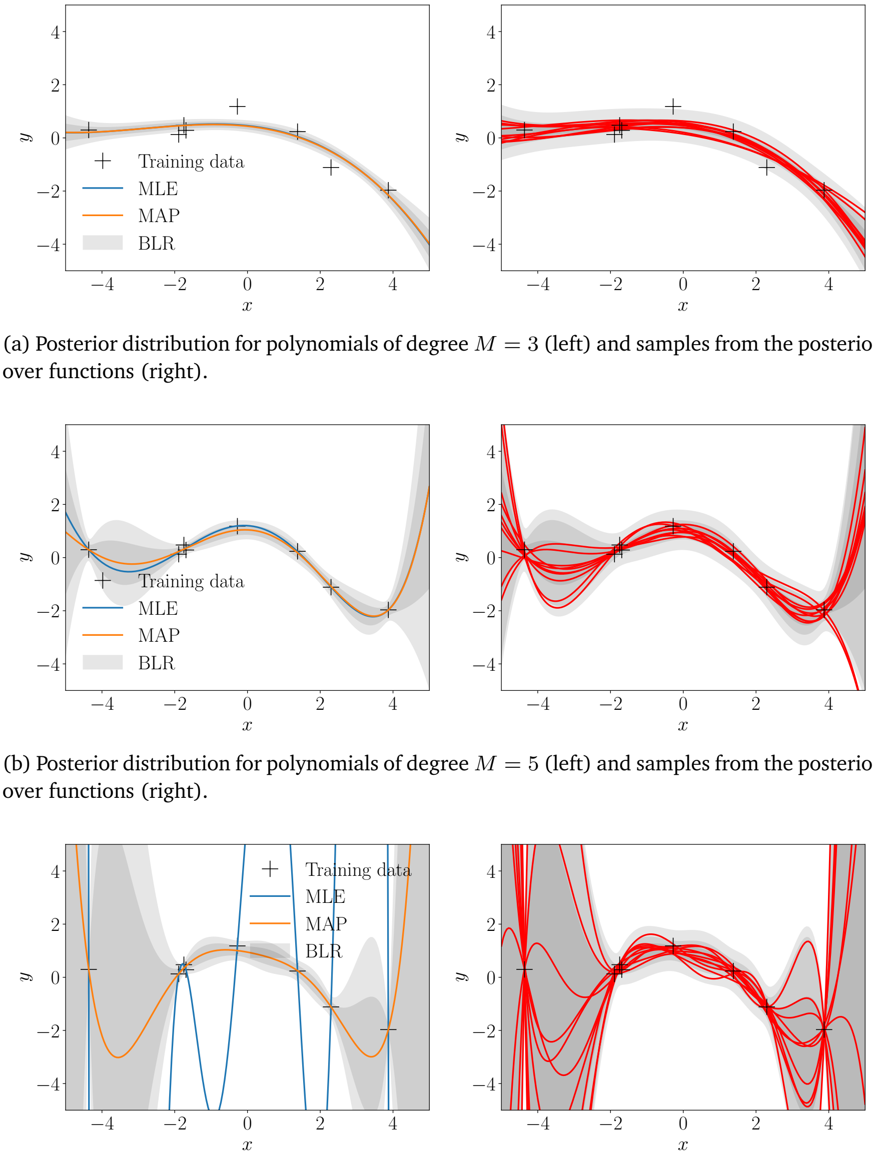
(a) Training data.

(b) Posterior over functions represented by the marginal uncertainties (shaded) showing the 95% predictive confidence bounds, the maximum likelihood estimate (MLE) and the MAP estimate (MAP), which is identical to the posterior mean function.

(c) Samples from the posterior over functions, which are induced by the samples from the parameter posterior.

Figure 9.10 Bayesian linear regression and posterior over functions. (a) Training data; (b) posterior distribution over functions; (c) Samples from the posterior over functions.

Figure 9.11 shows some posterior distributions over functions induced by the parameter posterior. For different polynomial degrees M the left panels show the maximum likelihood function $\boldsymbol{\theta}_{\mathrm{ML}}^{\top}\boldsymbol{\phi}(\cdot)$, the MAP function $\boldsymbol{\theta}_{\mathrm{MAP}}^{\top}\boldsymbol{\phi}(\cdot)$ (which is identical to the posterior mean function) and the 67% and 95% predictive confidence bounds obtained by Bayesian linear regression, represented by the shaded areas. The right panels show samples from the posterior over functions: Here, we sampled parameters $\boldsymbol{\theta}_i$ from the parameter posterior and computed the function $\boldsymbol{\phi}^{\top}(\boldsymbol{x}_*)\boldsymbol{\theta}_i$, which is a single realization of a function under the posterior distribution over functions. For low-order polynomials, the parameter posterior does not allow the parameters to vary much: The sampled functions are nearly identical. When we make the model more flexible by adding more parameters (i.e., we end up with a higher-order polynomial), these parameters are not sufficiently constrained by the posterior, and the sampled functions can be easily visually separated. We also see in the corresponding panels on the left how the uncertainty increases, especially at the boundaries. Although for a 7th-order polynomial the MAP estimate yields a reasonable fit, the Bayesian linear regression model additionally tells us that the posterior uncertainty is huge. This information can be critical when we use these predictions in a decision-making system, where bad decisions can have significant consequences (e.g., in reinforcement learning or robotics).



(a) Posterior distribution for polynomials of degree $M = 3$ (left) and samples from the posterior over functions (right).

(b) Posterior distribution for polynomials of degree $M = 5$ (left) and samples from the posterior over functions (right).

(c) Posterior distribution for polynomials of degree $M = 7$ (left) and samples from the posterior over functions (right).

Figure 9.11 Bayesian linear regression. Left panels: Shaded areas indicate the 67% (dark-gray) and 95% (light-gray) predictive confidence bounds. The mean of the Bayesian linear regression model coincides with the MAP estimate. The predictive uncertainty is the sum of the noise term and the posterior parameter uncertainty, which depends on the location of the test input. Right panels: Sampled functions from the posterior distribution.

9.3.5 Computing the Marginal Likelihood

In Section 8.5.2, we highlighted the importance of the marginal likelihood for Bayesian model selection. In the following, we compute the marginal likelihood for Bayesian linear regression with a conjugate Gaussian prior on the parameters, i.e., exactly the setting we have been discussing in this chapter.

Just to re-cap, we consider the following generative process:

$$\boldsymbol{\theta} \sim \mathcal{N}\big(\boldsymbol{m}_0,\, \boldsymbol{S}_0\big) \tag{9.60a}$$

$$y_n \,|\, \boldsymbol{x}_n, \boldsymbol{\theta} \sim \mathcal{N}\big(\boldsymbol{x}_n^\top \boldsymbol{\theta},\, \sigma^2\big)\,, \tag{9.60b}$$

The marginal likelihood can be interpreted as the expected likelihood under the prior, i.e., $\mathbb{E}_{\boldsymbol{\theta}}[p(\mathcal{Y}\,|\,\mathcal{X}, \boldsymbol{\theta})]$.

$n = 1, \dots, N$. The marginal likelihood is given by

$$\begin{aligned} p(\mathcal{Y}\,|\,\mathcal{X}) &= \int p(\mathcal{Y}\,|\,\mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} && (9.61a)\\ &= \int \mathcal{N}\big(\boldsymbol{y}\,|\,\boldsymbol{X}\boldsymbol{\theta},\, \sigma^2\boldsymbol{I}\big) \mathcal{N}\big(\boldsymbol{\theta}\,|\,\boldsymbol{m}_0,\, \boldsymbol{S}_0\big) \mathrm{d}\boldsymbol{\theta}\,, && (9.61b) \end{aligned}$$

where we integrate out the model parameters $\boldsymbol{\theta}$. We compute the marginal likelihood in two steps: First, we show that the marginal likelihood is Gaussian (as a distribution in $\boldsymbol{y}$); Second, we compute the mean and covariance of this Gaussian.

1 The marginal likelihood is Gaussian: From Section 6.5.2 we know that (i) the product of two Gaussian random variables is an (unnormalized) Gaussian distribution, (ii) a linear transformation of a Gaussian random variable is Gaussian distributed. In (9.61b), we require a linear transformation to bring $\mathcal{N}\big(\boldsymbol{y}\,|\,\boldsymbol{X}\boldsymbol{\theta},\, \sigma^2\boldsymbol{I}\big)$ into the form $\mathcal{N}\big(\boldsymbol{\theta}\,|\,\boldsymbol{\mu},\, \boldsymbol{\Sigma}\big)$ for some $\boldsymbol{\mu}, \boldsymbol{\Sigma}$. Once this is done, the integral can be solved in closed form. The result is the normalizing constant of the product of the two Gaussians. The normalizing constant itself has Gaussian shape, see (6.80).

2 Mean and covariance. We compute the mean and covariance matrix of the marginal likelihood by exploiting the standard results for means and covariances of affine transformations of random variables, see Section 6.4.4. The mean of the marginal likelihood is computed as

$$\mathbb{E}_{\boldsymbol{\theta}}[\mathcal{Y}\,|\,\mathcal{X}] = \mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}] = \boldsymbol{X}\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{\theta}] = \boldsymbol{X}\boldsymbol{m}_0\,. \tag{9.62}$$

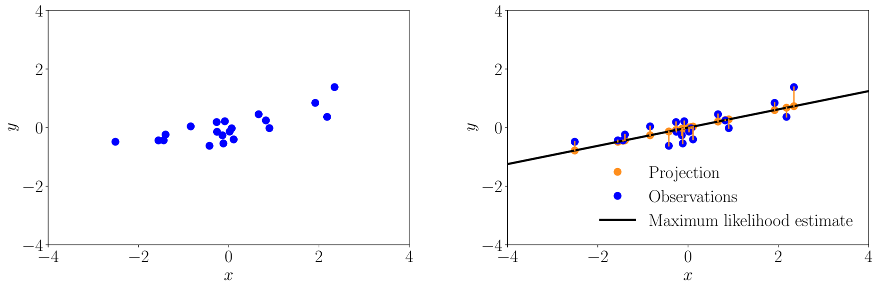
Note that $\boldsymbol{\epsilon} \sim \mathcal{N}\big(\boldsymbol{0},\, \sigma^2\boldsymbol{I}\big)$ is a vector of i.i.d. random variables. The covariance matrix is given as

$$\begin{aligned} \mathrm{Cov}_{\boldsymbol{\theta}}[\mathcal{Y}|\mathcal{X}] &= \mathrm{Cov}[\boldsymbol{X}\boldsymbol{\theta}] + \sigma^2\boldsymbol{I} = \boldsymbol{X}\,\mathrm{Cov}_{\boldsymbol{\theta}}[\boldsymbol{\theta}]\boldsymbol{X}^\top + \sigma^2\boldsymbol{I} && (9.63a)\\ &= \boldsymbol{X}\boldsymbol{S}_0\boldsymbol{X}^\top + \sigma^2\boldsymbol{I} && (9.63b) \end{aligned}$$

Hence, the marginal likelihood is

$$\begin{aligned} p(\mathcal{Y}\,|\,\mathcal{X}) &= (2\pi)^{-\frac{N}{2}} \det(\boldsymbol{X}\boldsymbol{S}_0\boldsymbol{X}^\top + \sigma^2\boldsymbol{I})^{-\frac{1}{2}} && (9.64a)\\ &\quad \cdot \exp\big(-\tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{m}_0)^\top(\boldsymbol{X}\boldsymbol{S}_0\boldsymbol{X}^\top + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{m}_0)\big) \\ &= \mathcal{N}\big(\boldsymbol{y}\,|\,\boldsymbol{X}\boldsymbol{m}_0,\, \boldsymbol{S}_0 + \sigma^2\boldsymbol{I}\big) && (9.64b) \end{aligned}$$

Given the close connection with the posterior predictive distribution (see Remark on page 308), the functional form of the marginal likelihood should not be too surprising.



(a) Regression dataset consisting of noisy observations y_n (blue) of function values $f(x_n)$ at input locations x_n.

(b) The orange dots are the projections of the noisy observations (blue dots) onto the line $\theta_{\text{ML}}x$. The maximum likelihood solution to a linear regression problem finds a subspace (line) onto which the overall projection error (orange lines) of the observations is minimized.

Figure 9.12 Geometric interpretation of least squares. (a) Dataset; (b) Maximum likelihood solution interpreted as a projection.

9.4 Maximum Likelihood as Orthogonal Projection

Having crunched through much algebra to derive maximum likelihood and MAP estimates, we will now provide a geometric interpretation of maximum likelihood estimation. Let us consider a simple linear regression setting

$$y = x\theta + \epsilon, \quad \epsilon \sim \mathcal{N}\big(0,\, \sigma^2\big)\,, \tag{9.65}$$

in which we consider linear functions $f : \mathbb{R} \to \mathbb{R}$ that go through the origin (we omit features here for clarity). The parameter θ determines the slope of the line. Figure 9.12(a) shows a one-dimensional dataset.

With a training data set $\{(x_1, y_1), \dots, (x_N, y_N)\}$ we recall the results from Section 9.2.1 and obtain the maximum likelihood estimator for the slope parameter as

$$\theta_{\text{ML}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} = \frac{\boldsymbol{X}^\top \boldsymbol{y}}{\boldsymbol{X}^\top \boldsymbol{X}} \in \mathbb{R}\,, \tag{9.66}$$

where $\boldsymbol{X} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$, $\boldsymbol{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$.

This means for the training inputs $\boldsymbol{X}$ we obtain the optimal (maximum likelihood) reconstruction of the training targets as

$$\boldsymbol{X}\theta_{\text{ML}} = \boldsymbol{X}\frac{\boldsymbol{X}^\top \boldsymbol{y}}{\boldsymbol{X}^\top \boldsymbol{X}} = \frac{\boldsymbol{X}\boldsymbol{X}^\top}{\boldsymbol{X}^\top \boldsymbol{X}}\boldsymbol{y}\,, \tag{9.67}$$

i.e., we obtain the approximation with the minimum least-squares error between $\boldsymbol{y}$ and $\boldsymbol{X}\theta$.

Linear regression can be thought of as a method for solving systems of linear equations.

As we are looking for a solution of $\boldsymbol{y} = \boldsymbol{X}\theta$, we can think of linear regression as a problem for solving systems of linear equations. Therefore, we can relate to concepts from linear algebra and analytic geometry that we discussed in Chapters 2 and 3. In particular, looking carefully at (9.67) we see that the maximum likelihood estimator θ_{ML} in our example from (9.65) effectively does an orthogonal projection of $\boldsymbol{y}$ onto the one-dimensional subspace spanned by $\boldsymbol{X}$. Recalling the results on orthogonal projections from Section 3.7, we identify $\frac{\boldsymbol{X}\boldsymbol{X}^\top}{\boldsymbol{X}^\top\boldsymbol{X}}$ as the projection matrix, θ_{ML} as the coordinates of the projection onto the one-dimensional subspace of $\mathbb{R}^N$ spanned by $\boldsymbol{X}$ and $\boldsymbol{X}\theta_{\text{ML}}$ as the orthogonal projection of $\boldsymbol{y}$ onto this subspace.

Maximum likelihood linear regression performs an orthogonal projection.

Therefore, the maximum likelihood solution provides also a geometrically optimal solution by finding the vectors in the subspace spanned by $\boldsymbol{X}$ that are "closest" to the corresponding observations $\boldsymbol{y}$, where "closest" means the smallest (squared) distance of the function values y_n to $x_n\theta$. This is achieved by orthogonal projections. Figure 9.12(b) shows the projection of the noisy observations onto the subspace that minimizes the squared distance between the original dataset and its projection (note that the x-coordinate is fixed), which corresponds to the maximum likelihood solution.

In the general linear regression case where

$$y = \boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{9.68}$$

with vector-valued features $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^K$, we again can interpret the maximum likelihood result

$$\boldsymbol{y} \approx \boldsymbol{\Phi}\boldsymbol{\theta}_{\text{ML}}\,, \tag{9.69}$$

$$\boldsymbol{\theta}_{\text{ML}} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \tag{9.70}$$

as a projection onto a K-dimensional subspace of $\mathbb{R}^N$, which is spanned by the columns of the feature matrix $\boldsymbol{\Phi}$, see Section 3.7.2.

If the feature functions ϕ_k that we use to construct the feature matrix $\boldsymbol{\Phi}$ are orthonormal (see Section 3.6), we obtain a special case where the columns of $\boldsymbol{\Phi}$ form an orthonormal basis (see Section 3.5), such that $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \boldsymbol{I}$. This will then lead to the projection

$$\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{y} = \left(\sum_{k=1}^{K}\boldsymbol{\phi}_k\boldsymbol{\phi}_k^\top\right)\boldsymbol{y} \tag{9.71}$$

so that the coupling between different features has disappeared and the maximum likelihood projection is simply the sum of projections of $\boldsymbol{y}$ onto the individual basis vectors $\boldsymbol{\phi}_k$, i.e., the columns of $\boldsymbol{\Phi}$. Many popular basis functions in signal processing, such as wavelets and Fourier bases, are orthogonal basis functions. When the basis is not orthogonal, one can convert a set of linearly independent basis functions to an orthogonal basis by using the Gram-Schmidt process (Strang, 2003).

9.5 Further Reading

In this chapter, we discussed linear regression for Gaussian likelihoods and conjugate Gaussian priors on the parameters of the model. This allowed for closed-form Bayesian inference. However, in some applications we may want to choose a different likelihood function. For example, in a binary *classification* setting, we observe only two possible (categorical) outcomes, and a Gaussian likelihood is inappropriate in this setting. Instead, we can choose a Bernoulli likelihood that will return a probability of the predicted label to be 1 (or 0). We refer to the books by Bishop (2006); Murphy (2012); Barber (2012) for an in-depth introduction to classification problems. A different example where non-Gaussian likelihoods are important is count data. Counts are non-negative integers, and in this case a Binomial or Poisson likelihood would be a better choice than a Gaussian. All these examples fall into the category of *generalized linear models*, a flexible generalization of linear regression that allows for response variables that have error distribution models other than a Gaussian distribution. The GLM generalizes linear regression by allowing the linear model to be related to the observed values via a smooth and invertible function $\sigma(\cdot)$ that may be nonlinear so that $y = \sigma(f)$, where $f = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$ is the linear regression model from (9.13). We can therefore think of a generalized linear model in terms of function composition $y = \sigma \circ f$ where f is a linear regression model and σ the activation function. Note, that although we are talking about "generalized linear models" the outputs y are no longer linear in the parameters $\boldsymbol{\theta}$. In *logistic regression*, we choose the *logistic sigmoid* $\sigma(f) = \frac{1}{1+\exp(-f)} \in [0, 1]$, which can be interpreted as the probability of observing $y = 1$ of a Bernoulli random variable $y \in \{0, 1\}$. The function $\sigma(\cdot)$ is called *transfer function* or *activation function*, its inverse is called the *canonical link function*. From this perspective, it is also clear that generalized linear models are the building blocks of (deep) feedforward neural networks: If we consider a generalized linear model $\boldsymbol{y} = \sigma(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{A}$ is a weight matrix and $\boldsymbol{b}$ a bias vector, we identify this generalized linear model as a single-layer neural network with activation function $\sigma(\cdot)$. We can now recursively compose these functions via

classification

generalized linear models

logistic regression

logistic sigmoid

transfer function

activation function

canonical link function

For ordinary linear regression the activation function would simply be the identity.

Generalized linear models are the building blocks of deep neural networks.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}_k(\boldsymbol{x}_k) \\ \boldsymbol{f}_k(\boldsymbol{x}_k) &= \sigma_k(\boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{b}_k) \end{aligned} \tag{9.72}$$

for $k = 0, \ldots, K-1$ where $\boldsymbol{x}_0$ are the input features and $\boldsymbol{x}_K = \boldsymbol{y}$ are the observed outputs, such that $\boldsymbol{f}_{K-1} \circ \cdots \circ \boldsymbol{f}_0$ is a K-layer deep neural network. Therefore, the building blocks of this deep neural network are the generalized linear models defined in (9.72). A great post on the relation between GLMs and deep networks is available at `https://tinyurl.com/glm-dnn`. Neural networks (Bishop, 1995; Goodfellow et al., 2016) are significantly more expressive and flexible than linear regression models. However, maximum likelihood parameter estimation is a non-convex optimization problem, and marginalization of the parameters in a fully Bayesian setting is analytically intractable.

Gaussian processes

We briefly hinted at the fact that a distribution over parameters induces a distribution over regression functions. *Gaussian processes* (Rasmussen and Williams, 2006) are regression models where the concept of a distribution over function is central. Instead of placing a distribution over parameters a Gaussian process places a distribution directly on the space of functions without the "detour" via the parameters. To do so, the Gaussian process exploits the *kernel trick* (Schölkopf and Smola, 2002), which allows us to compute inner products between two function values $f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)$ only by looking at the corresponding input $\boldsymbol{x}_i, \boldsymbol{x}_j$. A Gaussian process is closely related to both Bayesian linear regression and support vector regression but can also be interpreted as a Bayesian neural network with a single hidden layer where the number of units tends to infinity (Neal, 1996; Williams, 1997). An excellent introduction to Gaussian processes can be found in (MacKay, 1998; Rasmussen and Williams, 2006).

kernel trick

We focused on Gaussian parameter priors in the discussions in this chapters because they allow for closed-form inference in linear regression models. However, even in a regression setting with Gaussian likelihoods we may choose a non-Gaussian prior. Consider a setting where the inputs are $\boldsymbol{x} \in \mathbb{R}^D$ and our training set is small and of size $N \ll D$. This means that the regression problem is under-determined. In this case, we can choose a parameter prior that enforces sparsity, i.e., a prior that tries to set as many parameters to 0 as possible (*variable selection*). This prior provides a stronger regularizer than the Gaussian prior, which often leads to an increased prediction accuracy and interpretability of the model. The Laplace prior is one example that is frequently used for this purpose. A linear regression model with the Laplace prior on the parameters is equivalent to linear regression with L1 regularization (*LASSO*) (Tibshirani, 1996). The Laplace distribution is sharply peaked at zero (its first derivative is discontinuous) and it concentrates its probability mass closer to zero than the Gaussian distribution, which encourages parameters to be 0. Therefore, the non-zero parameters are relevant for the regression problem, which is the reason why we also speak of "variable selection".

variable selection

LASSO