Probabilistic machine learning

Karsten W. Jacobsen

Wednesday 6th January, 2021

Contents

Part I

Introduction

A number of excellent texts on probabilistic modeling and machine learning are available in the form of articles and text books, some of which are also accessible online.

- A short, broad overview of probabilistic machine learning can be found in [Gha15], which I highly recommend.
- An excellent standard textbook on machine learning is the one by Bishop [Bis06]. I have tried to keep my notation close to the one by Bishop, so this would be a good place to look up more information.
- The book by Trevor Hastie et al. [HTF09] is another "classic" textbook on statistical learning. It is freely available at `http://www.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf`.
- Sivia and Skilling have written a small book on data analysis [SS06]. This is an enjoyable Bayesian tutorial, but does not contain much machine learning.
- Ramussen and Williams have written a fairly advanced book on Gaussian processes [RW06]. This book is also available at the internet at `http://www.gaussianprocess.org/gpml/`.
- Finally, E. T. Jaynes has written a thoughtful book on probability theory with the ambitious subtitle *The Logic of Science* [Jay03]. Jaynes is one of the pioneers in understanding the basics of statistical physics.

Chapter 1

A very brief introduction to machine learning

There are many types of machine learning. If we have some data, which could be anything from information about how users choose music at Spotify to the atomic structure of materials for solar cells, one might ask questions about patterns or correlations in the data. This type of analysis is typically called *unsupervised*. The type of machine learning we shall be interested in here, is one where we have a dataset, and based on that, we want to make some predictions of new data. For example, we might know the composition and mass density of some materials and now we want to predict the mass densities of new materials based on their composition. This type of machine learning is called *supervised* learning.

Chapter 2

Fitting of a function

A simple example of supervised machine learning, which illustrates some of the challenges that we shall encounter, is the fitting of a function. Let us say that we have a function $f(x)$, where we only know the values $y_n = f(x_n)$ at certain points $x_n, n = 1, 2, \ldots, N$. Based on this information we want to predict the value $f(x)$ at another point x, which has not been investigated before. Figure 2.1 (left) illustrates this in a case where we know $N = 10$ points on a sine function: $y_n = \sin(2\pi * x_n)$. The model or fitting function we shall use, f_{fit}, is a polynomial

$$f_{\text{fit}}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1} \tag{2.1}$$

with M parameters. We shall see what is happening, when we vary $M = 1, 2, 3, 5, 10$. How do we determine the best polynomial? One way to do this is the so-called least-squares fit, where the value of $C = \frac{1}{2}\sum_{n=1}^{N}(y_n - f_{\text{fit}}(x_n))^2$ is minimized. C is called a *cost function* and it is minimized with respect to the model parameters $w_0, w_1, \ldots w_{M-1}$. We shall later derive the least-square method from probability considerations. Figure 2.1 (left) shows that if the order of the polynomial is low, say $M = 1, 2$ or 3, the fit does not at all go through the N points (note that the degree of the polynomial is $M - 1$). This phenomenon is called *bias*. If a model is too simple, *i.e.* has too few parameters, it cannot reproduce the data we have and this results in a

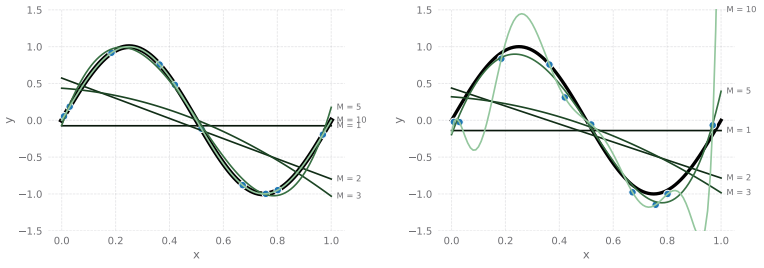

Figure 2.1: Left: Fitting a sine function (the black curve) with polynomials of different degree ($M - 1$). The nine'th order polynomial is seen to lie almost on top of the sine function. Right: Fitting a sine function (the black curve) with polynomials of different degree when the data are noisy.

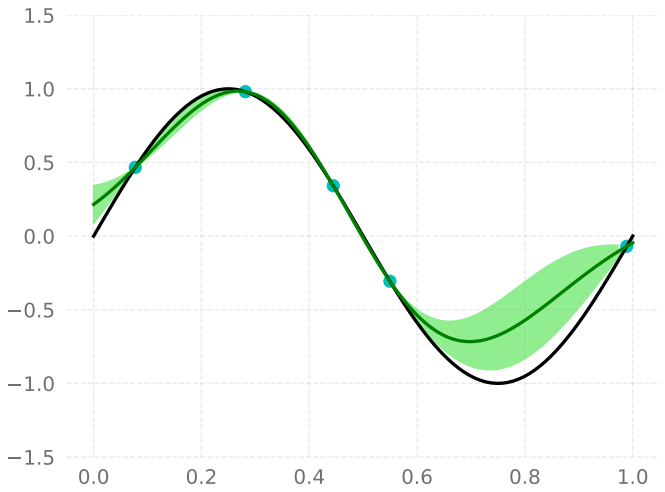

Figure 2.2: Fitting a sine function with a so-called Gaussian process, which provides both a fit and uncertainty estimates for the predictions (the light green area spans one standard deviation).

high bias.

The situation is better with higher degree polynomials, where the fit functions are much closer to the data points. (For $M = 10$ this is of course trivial since the function is the one and only polynomial that goes exactly through the points.) However, the quality of the higher-order fits change dramatically if the data are noisy. In Figure 2.1 (right) random noise with a typical value of ± 0.1 has been added to the data points so that they do not lie exactly on the sine function. Now the highest order polynomial still goes through all the data points (so it still has low bias), but it is a crazy approximation to the sine function in general. The phenomenon of fitting noise too closely leading to a poor overall model is called *overfitting*. It is clear that if we had picked some other data points or a different noise, the fit by the highest order polynomial would lead to completely different predictions. This is called high *variance* and happens when the model is too flexible, *i.e.* has too many parameters, which cannot be reliably determined by the data points. An ideal fit has both low bias and low variance, but the first one is obtained with an advanced model with many parameters and the second one with a simple model with few parameters, so there is a trade off between the two.

The simple example of fitting a sine function illustrates some of the challenges for machine learning: the complexity of a model has to be chosen with care to account for both bias and variance. When you are done with this note you will be much better at fitting. Figure 2.2 illustrates a fit based on a so-called Gaussian process that we shall explain in Section 17. The fit is based on only five points, (with ten points, the fit is so good that you cannot distinguish it from the sine function), and an estimate of the uncertainty of the prediction is also automatically available (the light green area). It is clear to see, that in the region far from the data points, the deviation between the predicted function and the sine function is large, but the model is ”aware” of this, since

the uncertainty is also large. Uncertainty estimation is founded on probability theory, and this is therefore what we shall look at next.

Part II

Probability theory

Chapter 3

The basics of probability theory

We can assign probabilities to *propositions*. If we throw a die, a proposition, Y, could for example be that we get the number 3, and if we believe the die to be fair, we would ascribe the probability $\mathcal{P}(Y) = 1/6$. A proposition of this kind is also called an *event*, *i.e.* the result of some experiment. However, we shall also ascribe probabilities to other types of propositions than events. For example, we shall ascribe probabilities to models: some models are better than others to describe a set of data correctly and they therefore have higher probabilities of being correct.

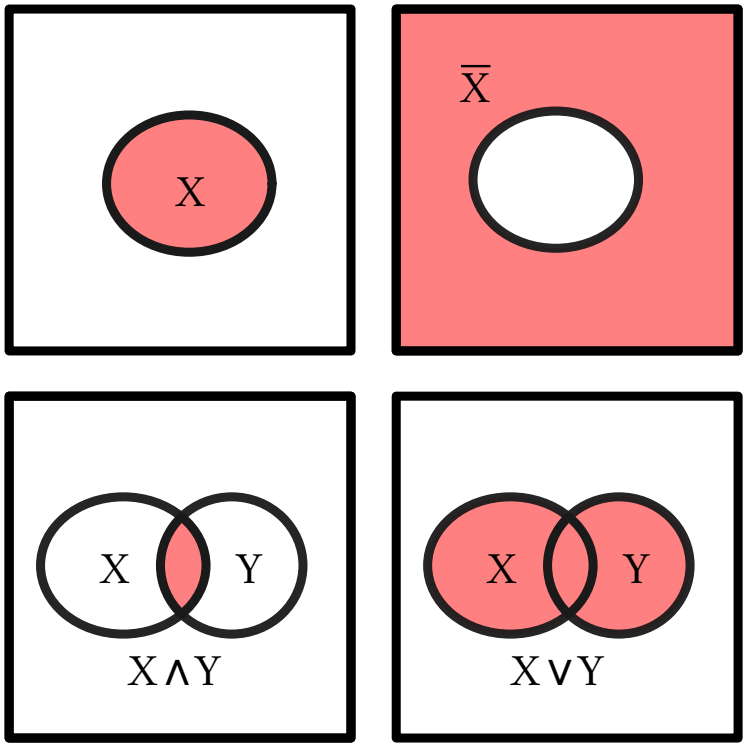

Figure 3.1: Visualization of events X, $\overline{X}$ (not X), $X \wedge Y$ (X and Y), and $X \vee Y$ (X or Y)

.

In the case of events, probabilities can be easily visualized as shown in Fig. 3.1. The area inside the square represent all possible events, and the "blobs" represent the events X and Y. The events $X \wedge Y$ (X and Y) and $X \vee Y$ (X or Y) are also shown.

Exercise: Boolean algebra

Show graphically, that $X \vee Y = \overline{\overline{X} \wedge \overline{Y}}$.
And that $\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$.
And $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$

Chapter 4

The sum and product rules

Probability theory is based on two fundamental rules, the sum rule and the product rule. The *sum rule* says that for two propositions X and I we have

$$\mathcal{P}(X|I) + \mathcal{P}(\overline{X}|I) = 1, \tag{4.1}$$

where $\mathcal{P}(X|I)$ means "probability of proposition X given I" (*i.e.* given that I is true). $\overline{X}$ denotes the negative proposition of X, so that $\overline{X}$ is true when X is false and vice versa. If we for example toss a die, the proposition X could be "we get 3", and then $\overline{X}$ would be "we get 1, 2, 4, 5, or 6". So basically the sum rule expresses that either X is true or $\overline{X}$ is true.
The formula looks even simpler, if we remove the proposition I: $\mathcal{P}(X) + \mathcal{P}(\overline{X}) = 1$. The role of the proposition I is to include all relevant "background" information. In the example with the die, the proposition I would include that we are in fact throwing a die.
The *product rule* says

$$\mathcal{P}(X, Y|I) = \mathcal{P}(X|Y, I)\mathcal{P}(Y|I). \tag{4.2}$$

Here $\mathcal{P}(X, Y|I)$ is the probability that both X and Y are true. This is the same as $\mathcal{P}(X \wedge Y|I)$, but is common to just use a comma between the two propositions. We have explicitly included the proposition I to indicate that there are always some previous assumptions that should be considered. Leaving out the I it looks even simpler: $\mathcal{P}(X, Y) = \mathcal{P}(X|Y)\mathcal{P}(Y)$.

Exercise: The product rule

Discuss the product rule graphically: The whole box is I, which we shall take to have an area of 1. The area of the blob labeled Y is $\mathcal{P}(Y|I)$. Continue the argument ...

Exercise: The extended sum rule

Show the *extended sum rule*

$$\mathcal{P}(X, Y|I) = \mathcal{P}(X|I) + \mathcal{P}(Y|I) - \mathcal{P}(X \vee Y|I), \tag{4.3}$$

where $\vee$ indicates the logical "or".

Exercise: Extended, extended sum rule

Use the extended sum rule to show, that if $Y_i, i = 1, 2, \ldots, n$ is a set of mutually exclusive propositions (*i.e.* that $\mathcal{P}(Y_i, Y_j) = 0$ for $i \neq j$), which together cover all possible propositions included in I, we have

$$\sum_{i=1}^{n} \mathcal{P}(Y_i|I) = 1. \tag{4.4}$$

We shall also sometimes write this as $\sum_Y \mathcal{P}(Y|I) = 1$, or if Y is continuous

$$\int \mathcal{P}(Y|I)\, dY = 1. \tag{4.5}$$

A generalization of this is

$$\sum_Y \mathcal{P}(X, Y|I) = \mathcal{P}(X|I) \;\text{ or }\; \int \mathcal{P}(X, Y|I)\, dY = \mathcal{P}(X|I) \tag{4.6}$$

as can be shown using the product rule.
This formula can be combined with the product rule to

$$\mathcal{P}(X|I) = \int \mathcal{P}(X, Y|I)\, dY = \int \mathcal{P}(X|Y, I)\mathcal{P}(Y|I)\, dY. \tag{4.7}$$

This way of "removing" the parameter Y is called *marginalization*. (In physics it is often called to *integrate out* a parameter).
If the knowledge about proposition Y does not affect the probability of proposition X the two propositions are called *independent*. The requirement of independence can be formulated

$$\mathcal{P}(X|I) = \mathcal{P}(X|Y, I). \tag{4.8}$$

It the then follows from the product rule that

$$\mathcal{P}(X, Y|I) = \mathcal{P}(X|Y, I)\mathcal{P}(Y|I) = \mathcal{P}(X|I)\mathcal{P}(Y|I). \tag{4.9}$$

Exercise: Independence of propositions

Show that if X and Y are independent propositions then

$$\mathcal{P}(X \vee Y|I) = \mathcal{P}(X|I) + \mathcal{P}(Y|I) - \mathcal{P}(X|I)\mathcal{P}(Y|I) \tag{4.10}$$

Chapter 5

Bayes' theorem

An important theorem that we shall use in many different contexts is *Bayes' theorem*. It reads,

$$\mathcal{P}(Y|X,I) = \frac{\mathcal{P}(X|Y,I)\mathcal{P}(Y|I)}{\mathcal{P}(X|I)}. \tag{5.1}$$

This is easily shown from the fact that $\mathcal{P}(X,Y|I) = \mathcal{P}(Y,X|I)$ by applying the product rule. Leaving out I it looks simpler $\mathcal{P}(Y|X) = \mathcal{P}(X|Y)\mathcal{P}(Y)/\mathcal{P}(X)$, but the version including I is more general.

Exercise: Girls or boys

Use Bayes' theorem to analyze the following problem:
Alice and Bob have two children. What is the probability that they have two girls, when you get the following information (four different cases)?

1. They have at least one girl ("$\geq$1g")
2. The oldest child is a girl ("old")
3. They have a girl with blue eyes ("blue")
4. They have a girl with the name "Mushroom" ("Mushroom")

Exercise: Medical screening

The successful treatment of diseases may depend critically on early discovery, and in some cases large-scale screenings for diseases like breast cancer have been introduced. Since clinical tests are seldomly perfect, there is a risk that some people will be upset by getting a positive test (meaning the test shows them to be ill), while they are actually perfectly healthy. False negatives, where the test shows a person to be healthy, but the person is actually ill, may of course also be problematic.
We shall analyze this situation using Bayes' theorem. Consider a test, which is 99% good, meaning that if a person is ill, the test will be positive in 99% of the cases and, likewise, if the person is healthy, the test will be negative in 99% of those cases. Mathematically, this can be expressed $\mathcal{P}(negative|healthy) = \mathcal{P}(positive|ill) = 0.99$, or by using abbreviations $\mathcal{P}(neg|h) = \mathcal{P}(pos|ill) =$

Figure 5.1: A mobile clinic used to screen coal miners at risk of black lung disease. Source:https://en.wikipedia.org/wiki/Screening_(medicine)#/media/File:NIOSH_Mobile_Health_Screenings_(16027817612).jpg

0.99. Since we only have two outcomes of the test (positive or negative), the sum rule immediately gives us $\mathcal{P}(pos|h) = \mathcal{P}(neg|ill) = 0.01$.
Now, a patient comes to the doctor and gets a test. Unfortunately, the test is positive. What is the probability, that the patient is ill? Use Bayes' theorem to find this probability $\mathcal{P}(ill|pos)$, assuming that it is a fairly rare disease that only 0.1% of the population has ($\mathcal{P}(ill) = 0.001$). You shall need that $\mathcal{P}(pos) = \mathcal{P}(pos, ill) + \mathcal{P}(pos, h) = \mathcal{P}(pos|ill)\mathcal{P}(ill) + \mathcal{P}(pos|h)\mathcal{P}(h)$, which follows from Eq. (4.6). Discuss, how the result depends on the frequency of the disease in the population.

Chapter 6

Models and data

Bayes' theorem can also be used when considering models ($\mathcal{M}$) and data ($\mathcal{D}$). In that case it looks like

$$\mathcal{P}(\mathcal{M}|\mathcal{D}) = \frac{1}{\mathcal{P}(\mathcal{D})}\mathcal{P}(\mathcal{D}|\mathcal{M})\mathcal{P}(\mathcal{M}) \tag{6.1}$$

The term $\mathcal{P}(\mathcal{M})$ is called the *prior* probability of the model. This is the probability before any data are taken into account. $\mathcal{P}(\mathcal{D}|\mathcal{M})$ is the *likelihood*. It tells whether the data are likely if we assume that the model is true. The end result $\mathcal{P}(\mathcal{M}|\mathcal{D})$ is the probability for the model after the data are considered and is called the *posterior* probability.
The denominator $\mathcal{P}(\mathcal{D})$ is a normalization constant, that we often do not have to calculate explicitly. It ensures that if we sum over all possible models, we should end up getting one. Using the sum and product rules it can be written

$$\mathcal{P}(\mathcal{D}) = \mathcal{P}(\mathcal{D},\mathcal{M}) + \mathcal{P}(\mathcal{D},\bar{\mathcal{M}}) = \mathcal{P}(\mathcal{D}|\mathcal{M})\mathcal{P}(\mathcal{M}) + \mathcal{P}(\mathcal{D}|\bar{\mathcal{M}})\mathcal{P}(\bar{\mathcal{M}}). \tag{6.2}$$

Exercise: Newton's second law

Why do we believe Newton's second law? Discuss this in light of Bayes' theorem. First show, that Bayes' theorem can be written

$$\mathcal{P}(\mathcal{M}|\mathcal{D}) = \frac{\dfrac{\mathcal{P}(\mathcal{D}|\mathcal{M})}{\mathcal{P}(\mathcal{D}|\bar{\mathcal{M}})}\cdot\dfrac{\mathcal{P}(\mathcal{M})}{\mathcal{P}(\bar{\mathcal{M}})}}{1+\dfrac{\mathcal{P}(\mathcal{D}|\mathcal{M})}{\mathcal{P}(\mathcal{D}|\bar{\mathcal{M}})}\cdot\dfrac{\mathcal{P}(\mathcal{M})}{\mathcal{P}(\bar{\mathcal{M}})}}, \tag{6.3}$$

and then discuss the expression.

One way of viewing Bayes' theorem is that it expresses an update of our probability of the model based on information, *i.e.* data. This update can be an iterative process with sequential steps. To see this we write Bayes' theorem in the form

$$\mathcal{P}(\mathcal{M}|\mathcal{D}_{new},\mathcal{D}_{old}) = \frac{1}{\mathcal{P}(\mathcal{D}_{new}|\mathcal{D}_{old})}\mathcal{P}(\mathcal{D}_{new}|\mathcal{M},\mathcal{D}_{old})\mathcal{P}(\mathcal{M}|\mathcal{D}_{old}). \tag{6.4}$$

We see that here $\mathcal{P}(\mathcal{M}|\mathcal{D}_{old})$ plays the role of the prior with respect to the new data and that the end result is the probability for the model taking into account both the new and the old data.

Many models involve a set of parameters, $\boldsymbol{w}$, just like our polynomial fits above in Section 1. Having settled on a particular class of models (say, the polynomials) we therefore want to know the probability distribution for the parameters. The "best" fit function can then for example be identified as the one with the highest probability. Again Bayes' theorem comes to help in the form

$$\mathcal{P}(\boldsymbol{w}|\mathcal{D},\mathcal{M}) = \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w},\mathcal{M})}{\mathcal{P}(\mathcal{D},\mathcal{M})} = \frac{\mathcal{P}(\mathcal{D}|\boldsymbol{w},\mathcal{M})\mathcal{P}(\boldsymbol{w}|\mathcal{M})}{\int \mathcal{P}(\mathcal{D}|\boldsymbol{w}',\mathcal{M})\mathcal{P}(\boldsymbol{w}'|\mathcal{M})d\boldsymbol{w}'}, \quad (6.5)$$

where we in the last equation have removed a factor $\mathcal{P}(\mathcal{M})$ in both the numerator and the denominator. The factor $\mathcal{P}(\boldsymbol{w}|\mathcal{M})$ now works as the prior probability distribution for the parameters given that we have decided on the model $\mathcal{M}$.

Exercise: Posterior parameter distribution

Show Eq. (6.5).

Chapter 7

Model selection

Finally, we can also use Bayes' theorem to compare two different models, $\mathcal{M}_1$ and $\mathcal{M}_2$, with separate sets of parameters, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. Given some data, $\mathcal{D}$, the ratio of probabilities for the two models are (from Eq. (6.1))

$$\frac{\mathcal{P}(\mathcal{M}_2|\mathcal{D})}{\mathcal{P}(\mathcal{M}_1|\mathcal{D})} = \frac{\mathcal{P}(\mathcal{D}|\mathcal{M}_2)}{\mathcal{P}(\mathcal{D}|\mathcal{M}_1)} \cdot \frac{\mathcal{P}(\mathcal{M}_2)}{\mathcal{P}(\mathcal{M}_1)} \tag{7.1}$$

$$= \frac{\int \mathcal{P}(\mathcal{D}|\mathcal{M}_2, \boldsymbol{w}_2)\mathcal{P}(\boldsymbol{w}_2|\mathcal{M}_2)d\boldsymbol{w}_2}{\int \mathcal{P}(\mathcal{D}|\mathcal{M}_1, \boldsymbol{w}_1)\mathcal{P}(\boldsymbol{w}_1|\mathcal{M}_1)d\boldsymbol{w}_1} \cdot \frac{\mathcal{P}(\mathcal{M}_2)}{\mathcal{P}(\mathcal{M}_1)} \tag{7.2}$$

The first factor on the right hand side involves the likelihoods, while the second factor contains the prior probabilities for the models. Eq. 7.1 can be used to select the best model, *i.e.* the model with the highest probability. We saw in the introductory section about fitting, that including more parameters in a model will always lead to a better fit in the sense that the model is better at reproducing the known data. However, as we shall see later, Bayes' theorem subscribes to the principle of Occams' razor. Occam's razor says that if you have two explanations for a phenomenon, the one with the smallest number of assumptions is usually the correct one. This is closely related to the quote "Everything should be as simple as it can be, but not simpler", which is sometimes ascribed to Einstein.
In a machine learning context, this principle favors simple models with few parameters over more complicated ones. As we shall see later the likelihood $\mathcal{P}(\mathcal{D}|\mathcal{M})$ automatically includes this aspect.

Chapter 8

Probability distributions

Here we define some basic quantities related to probability distributions or probability densities as they may be called when they are functions of continuous variables. Let $\mathcal{P}(x)$ denote a function of the variable x. For $\mathcal{P}$ to be interpreted as a probability density we need $\mathcal{P}(x) \geq 0$ everywhere and $\int P(x)\,dx = 1$. The average or expectation value $\langle f \rangle$ of a function $f(x)$ is then defined as

$$\langle f \rangle = \int f(x)\mathcal{P}(x)\,dx, \tag{8.1}$$

and the variance is

$$\text{Var}[f] = \langle (f - \langle f \rangle)^2 \rangle = \langle f^2 \rangle - \langle f \rangle^2. \tag{8.2}$$

In the simple but important case where $f(x) = x$, we get

$$\text{Var}[x] = \langle x^2 \rangle - \langle x \rangle^2 \tag{8.3}$$

Many of the probability distributions we are going to consider depend on several variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$. We shall consider $\boldsymbol{x}$ as a column vector and the transposed vector $\boldsymbol{x}^T$ as the corresponding row vector. The expression $\boldsymbol{x}^T\boldsymbol{x}$ is therefore a scalar, while $\boldsymbol{x}\boldsymbol{x}^T$ is a $n \times n$ matrix. We can then define the covariance matrix

$$\text{Cov}[\boldsymbol{x}, \boldsymbol{x}^T] = \langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle)(\boldsymbol{x}^T - \langle \boldsymbol{x}^T \rangle) \rangle = \langle \boldsymbol{x}\boldsymbol{x}^T \rangle - \langle \boldsymbol{x} \rangle \langle \boldsymbol{x} \rangle^T \tag{8.4}$$

Given a probability distribution $\mathcal{P}(x)$ on can imagine "pulling" points $x_1, x_2, \ldots, x_N$ from the distribution. Such a collection of points is called an *ensemble*. This concept is used a lot in statistical physics where for example a system at a particular temperature is represented by copies or "snapshots" of the system with different atomic coordinates and velocities. An ensemble makes it possible to approximate the calculation of averages like

$$\langle f \rangle = \int f(x)\mathcal{P}(x)\,dx \sim \frac{1}{N}\sum_{n=1}^{N} f(x_n). \tag{8.5}$$

Because the density of points in the ensemble already represent the probability density, the average appears as a plain average over the ensemble in the last expression.

Chapter 9

The Gaussian distribution

There are many different probability distributions, which have been carefully characterized, and where for example the average values and variances are known. See for example "List of probability distributions" at Wikipedia.
The probability distribution that we shall use the most is the Gaussian or normal distribution, which we write as

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(x-\mu)^2/2\sigma^2} \tag{9.1}$$

It is straightforward to show that for the Gaussian distribution $\langle x\rangle = \mu$ and Var$[x] = \sigma^2$.
The Gaussian distribution appears in many different contexts. If we have a stochastic variable, y, which is the sum of many different variables $y = y_1 + y_2 + \ldots$ with independent distributions $\mathcal{P}(y_i)$ then one can show that under certain assumptions the distribution $\mathcal{P}(y)$ becomes a Gaussian. This is called the "central limit theorem". You can look it up at Wikipedia.
To illustrate this, we consider the binomial distribution, which describes a series of heads/tails experiments with a probability p. It is given by

$$\mathcal{P}(n|N,p) = \binom{N}{n} p^n (1-p)^{(N-n)}, \tag{9.2}$$

where N is the total number of experiments and n is the number of heads (see more in the appendix). The average and the variance are given by $\langle n\rangle = pN$ and Var$[n] = Np(1-p)$. Since the binomial distribution is constructed from a series of independent events, it fulfills the criteria for the central limit theorem, so for large N it can be approximated by a Gaussian. Arranging that the Gaussian has the same average and variance as the binomial we get

$$\mathcal{P}(n|N,p) \sim \frac{1}{N}\mathcal{N}(x = n/N|p, p(1-p)/N), \tag{9.3}$$

where the factor $1/N$ in front ensures the right normalization, so that the Gaussian is normalized as a function of x.
The Gaussian distribution has a number of nice properties. The product of two Gaussians is for example a new Gaussian:

$$\mathcal{N}(x|\mu,\sigma^2) \propto \mathcal{N}(x|\mu_1,\sigma_1^2)\cdot\mathcal{N}(x|\mu_2,\sigma_2^2), \tag{9.4}$$

where the average value and variance of the new Gaussian are given by

$$\mu = \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right)\sigma^2 \text{ and } \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}. \tag{9.5}$$

The normalization constant in Eq. (9.4) is rarely needed.
This product rule is useful in cases where both the prior and the likelihood are Gaussian distributions, because then the posterior distribution also becomes a Gaussian. More generally, it is convenient if the prior and the posterior are given by the same type of probability distribution. Then the idea of updating the probability distribution by adding more data works easily. For a given likelihood, the probability distribution which has this property is called the *conjugate prior*. The Gaussian distribution is thus its own conjugate prior. You can read more about conjugate priors at Wikipedia including a list of conjugate priors for different probability distributions.

Exercise: From maximum entropy to the Gaussian distribution (optional)

In this problem you will need to know optimization under constraints and functional derivatives. If you don't, then follow the course on Advanced Quantum Mechanics, where I teach this!
One way to obtain the Gaussian distribution is from maximum entropy. The entropy S for a probability distribution is defined as

$$S = -\int \mathcal{P}(x)\log(\mathcal{P}(x))\,dx \tag{9.6}$$

Show that maximizing the entropy under the constraints $\langle x\rangle = \mu$ and $\text{Var}[x] = \sigma^2$ leads to a Gaussian distribution.

Chapter 10

Applications of Bayes' theorem

Exercise: Gaussianly distributed data

Let us consider a set of data $x_1, x_2, \ldots, x_N$, that we assume are Gaussianly distributed (this is our "model"), *i.e.* $\mathcal{P}(x_1, x_2, \ldots x_N|\mu, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \sigma^2)$. We shall assume that we know the variance σ^2, and are interested in finding the probability distribution for μ given the data: $\mathcal{P}(\mu|x_1, x_2, \ldots x_N, \sigma^2)$. We therefore use Bayes' theorem in the form

$$\mathcal{P}(\mu|x_1, x_2, \ldots x_N, \sigma^2) \propto \mathcal{P}(x_1, x_2, \ldots x_N|\mu, \sigma^2)\mathcal{P}(\mu|\sigma^2). \tag{10.1}$$

We first have to decide on a prior distribution $\mathcal{P}(\mu|\sigma^2)$. We have no particular information about μ before we look at the data, so we shall just take $\mathcal{P}(\mu|\sigma^2) = 1$.

This means that the posterior distribution equals the likelihood. Determine the posterior distribution and show that it can be expressed in terms of the following two quantities alone:

$$\bar{x} = \frac{1}{N}\sum_i x_i \text{ and } \overline{\Delta x^2} = \frac{1}{N}\sum_i (x_i - \bar{x})^2. \tag{10.2}$$

Such quantities that include all the relevant information about the data are sometimes called *sufficient statistics*.

Find the most probable values of μ by maximizing $\mathcal{P}(\mu|x_1, x_2, \ldots x_N, \sigma^2)$ (it is easiest to maximize $\log(\mathcal{P})$). How well do we determine the average, *i.e.* what is the variance of μ? What happens to this variance as we get more data points?

Exercise: Tversky and Kahneman

Amos Tversky and Daniel Kahneman are (mathematical) psychologists who worked on the psychology of prediction, probability judgment, and decision-making. Kahneman received the Nobel prize in economics in 2002. (Tversky had died at that time.) In part of their work they demonstrate that humans tend to make decisions based on wrong ideas about probabilities.

In their article "Judgement under uncertainty - heuristics and biases" in Science in 1974 they have asked the following question to a number of undergraduates:

A certain town is served by two hospitals. In the larger hospital about 45 babies are born each day, and in the smaller hospital about 15 babies are born each day. As you know, about 50 percent of all babies are boys. However, the exact percentage varies from day to day. Sometimes it may be higher than 50 percent, sometimes lower.
For a period of 1 year, each hospital recorded the days on which more than 60 percent of the babies born were boys. Which hospital do you think recorded more such days?

- *The larger hospital (21)*
- *The smaller hospital (21)*
- *About the same (that is, within 5 percent of each other) (53)*

The values in parentheses are the number of undergraduate students who chose each answer.
What is the correct answer?

Exercise: Playing with priors – the average height of Norwegians

Let us say, that we have measured the heights $h_1, h_2 \ldots h_N$ of N Norwegian men. What is then the best estimate of the average height h_{Nor} of Norwegian men given that we know that the average height of Danish men is $h_{\text{DK}} = 181.4\,\text{cm}$? How does the height of Danish men come into this, you might ask. Isn't the best estimate simply

$$h_{\text{Nor}} = \frac{1}{N}\sum_i h_i \tag{10.3}$$

Well, let us see if we can do better.

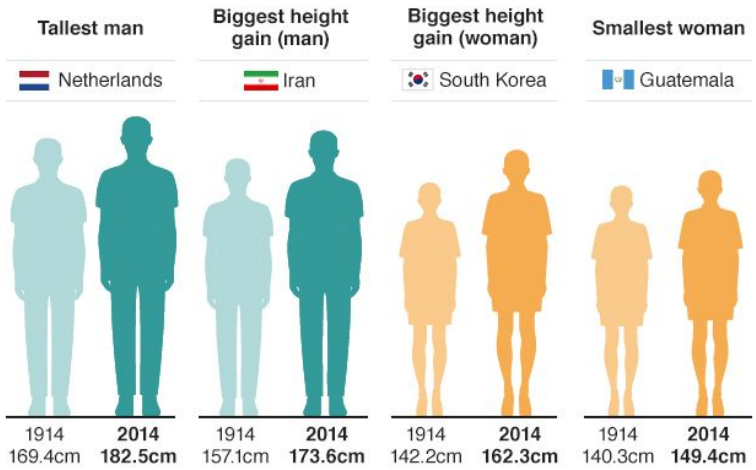

Figure 10.1: What is the average height of Norwegian men? (Figure from bbc.com)

First we need a model for the distribution of the heights of Norwegian men and we shall (of course) take this as a Gaussian $\mathcal{N}(h|h_{\text{Nor}}, \sigma^2_{\text{Nor}})$. What is a

reasonable value for the width σ_{Nor}? In principle we could try to estimate this from the data, but right now we shall just assume the value $\sigma_{\text{Nor}} = 5\,\text{cm}$. This roughly means that 2/3 of all Norwegian men have their heights within the average plus/minus $5\,\text{cm}$, which sounds reasonable.

How does the height of Danish men come into play? Well, we are allowed to use a prior probability distribution. (In fact, we are not only allowed to do that, it is simply required by Bayes' theorem – there is no way around it). A simple choice could be $P_0 = 1$, and then we know from the example above that the best estimate would indeed be Eq. 10.3.

There is reason to assume that the average height of Norwegian men is not that different from the average height of Danish men. The tallest average height in the world is for Dutch men, and it is $182.5\,\text{cm}$, only about $1\,\text{cm}$ higher than for the Danes. So it is very likely that the average height of Norwegian men lies within, say, $\sigma_d = 2.5\,\text{cm}$ of the Danish one. Using again the Gaussian distribution, this means that our prior for the Norwegian average is $\mathcal{N}(h_{\text{Nor}}|h_{\text{DK}}, \sigma_d^2)$.

Determine the posterior probability distribution for the Norwegian average height. What is the value with the highest probability? Show that the result can be written as in Eq. (10.3) but now with a number (which number?) of Danish men of average height included in the mean. How confident can we be about our estimate? What is the uncertainty?

Exercise: Is the coin fair?

A friend and you have to find out who should wash the dishes. The obvious way to decide this is to flip a coin, and luckily your friend has a coin in her pocket. However, before you enter into this crucial game, you would like to know if the coin is fair with an equal chance of showing heads and tails. You therefore decide to test the coin (of course hoping that your friend will be impatient and wash the dishes in the meantime). You flip the coin N times out of which you get N_h heads (and $N_t = N - N_h$ tails). Do you trust the coin? What can you conclude if you get 6 heads out of 10 experiments? What if you get 600 heads out of 1000 experiments? (At this point your friend definitely either did the dishes or left). Analyze the situation using Bayes' theorem.

A bit of help: We assume that the result of the experiment can be described by a binomial distribution with probability p for heads. This is our model, which is described by the single variable p. We shall assume that the prior distribution is $\mathcal{P}_0(p) = 1$. (This is maybe not very nice to your friend, exhibiting a clear amount of distrust. You could also try a prior distribution, which is peaked around $p = 1/2$, but $\mathcal{P}_0(p) = 1$ is easier to work with.) The posterior distribution over p becomes a beta distribution $\mathcal{B}(p|\alpha, \beta)$, which is the conjugate of the binomial distribution. You can see the most basic information about the beta distribution in the appendix and find a lot more at Wikipedia.

Plot the posterior distribution for different counts of heads and tails. Calculate, analytically, the average value of p with the posterior distribution. The prior gives rise to addition of some "pseudo" counts for the coin, when calculating the average. How many pseudo heads are added and how many pseudo tails? The average is our best guess at the value of p. What is the uncertainty? You

Figure 10.2: DUBLIN (Sept. 1, 2012) U.S. Ambassador to Ireland, the Honorable Daniel Rooney (center left) and Irish Prime Minister Enda Kenny take part in the opening coin toss during the NCAA Emerald Isle Classic college football season opener in Aviva Stadium. Notre Dame played Navy for the 86th straight year, making it the longest continuous intersectional rivalry in the U.S. (U.S. Navy photo by Senior Chief Mass Communication Specialist Michael Lewis)

throw the coin 10 times and get 6 heads. Is the coin fair? What if you get 600 tails out of 1000 experiments?

Maybe you trust your friend more than expressed by $\mathcal{P}_0(p) = 1$. You might have flipped one of your friend's coins many times before, and it has always seemed fair. You therefore decide to assume at the outset that the coin is reasonably fair. Because the beta distribution is the conjugate of the binomial distribution it is easiest to use a beta distribution as the prior. The beta distribution $\mathcal{B}(p|\alpha, \alpha)$ peaks more and more around 1/2 in a symmetric way, when α is increased. (Plot it to see what it looks like). Calculate a new posterior distribution for p with a symmetric beta prior. What is now the best estimate of p? And the uncertainty? Can we again interpret this in terms of pseudo counts?

Remember one thing: There is no such thing as "no prior". If you decide to "neglect" the prior, it really means that you take $\mathcal{P}_0(p) = 1$, which as we have seen expresses some degree of mistrust for your friend. You cannot escape making a decision for the prior.

Exercise: But *is* the coin really fair? (optional)

In the previous problem we found the posterior probability distribution for p and from that we got an idea about how fair the coin is. But there is another way. As described in Section 7, Bayes' theorem can be used to select between different models, and we shall therefore consider the following two models:

$\mathcal{M}_1$ This is the model used above, where we regard the data as being binomially distributed with a probability parameter p.

$\mathcal{M}_2$ In this model we shall simply assume that the coin is fair, *i.e.* the data is assumed to be binomially distributed with a probability parameter of 1/2.

It might seem that model $\mathcal{M}_2$ is just a special case of $\mathcal{M}_1$, however, there is a difference: $\mathcal{M}_1$ contains a parameter (p), while $\mathcal{M}_2$ has no parameters. In that sense model $\mathcal{M}_2$ is simpler, and Bayes' theorem will have to make a balance between reproducing data, which favors model $\mathcal{M}_1$ and simplicity (Occam's razor!), which favors model $\mathcal{M}_2$.
Calculate the relative probability $\mathcal{P}(\mathcal{M}_2|\mathcal{D})/\mathcal{P}(\mathcal{M}_1|\mathcal{D})$ using Eq. (7.1), and discuss when the coin is fair (*i.e.* the simple model $\mathcal{M}_2$ is preferred).

Exercise: Course evaluations at DTU (optional)

(This problem is similar in spirit to the coin problem above, but more difficult. You can skip it if you do not have the guts.)

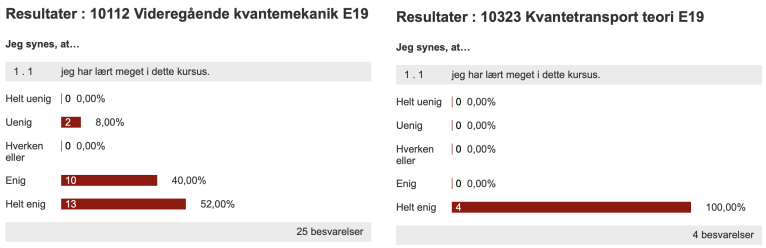

Figure 10.3: Student evaluations of two courses at DTU, 2019

Students at DTU every semester evaluate the courses they follow on a scale from 1 to 5, with 5 being the best. Let us say that for a given course, there are N_n students that give the grade n. The traditional average for the grade, g, is then $\langle g \rangle = (\sum_n nN_n)/N$, where $N = \sum_n N_n$ is the number of students participating in the evaluation. (The number of students participating in the evaluations is often far too small compared to the number of students taking the course, but that is a different story that we shall not go into here). Using this estimate, we can see that the grade for the course on Quantum Transport Theory is a straight 5 based on 4 student evaluations (see the figure). The variance of the data is also zero, so that seems to indicate that the grade is 5 with a very high degree of certainty. Is this a fair evaluation of the quality of the course (as seen by the students)? What if more students had participated in the course? Would they all give the grade 5? This would be quite unusual. Very few courses, if any, ends up with only 5's if there are more than 10 students. So, can we make a better estimate of the grade for the course and the uncertainty?
A bit more help: The model here consists of an assumption of five probabilities $p_n, n = 1, 2, 3, 4, 5$ for the grades n. The final grade is then given by $g = \sum_n np_n$. The simplest prior just assumes that the sum of the five probabilities is one: $\mathcal{P}_0(p_1, p_2, p_3, p_4, p_5) = \delta(1 - \sum_n p_n)$. The likelihood becomes

a multinomial distribution in the counts and the posterior becomes a Dirichlet distribution in the probabilities (Look it up – Wikipedia is great).

Part III

Machine learning

Equipped with Bayes' theorem, we are ready to take up the topic of machine learning again. We shall look at concepts like bias and variance when fitting a function, but we can now approach this in a probabilistic way. The first thing we shall do is to derive the least-squares method from an assumption about Gaussian noise.

A bit about notation: we shall encounter quite a few vectors and matrices, and I shall try to keep as close as possible to the following notation: Data points will be labeled by $n, m, \ldots$ as in $\boldsymbol{x}_n$ and the number of data points will often be denoted by N. Maybe you already noted that a data point $\boldsymbol{x}_n$ is in bold face. This is because each data point may be a vector of a certain length. In the simple example with the polynomial fit we only had a single variable, but we might be interested in the more general case with D (for dimension) components. We shall usually not need to address the individual components. The model parameters (*i.e.*, the coefficients w_i in the polynomial fit) will be labeled by $i, j, \ldots$ and the number of parameters is usually denoted by M.

Chapter 11

Method of least squares

Consider a function $f(x)$, that we imagine we "measure" at a given set of x-values $x_1, x_2, \ldots x_N$. However, the measured values are not given by just $f(x_n), i = 1, 2, \ldots, N$, because the experiments involve a certain degree of noise, which we shall assume is Gaussian. In other words, we assume the measured values $\boldsymbol{t} = (t_1, t_2, \ldots, t_N)$ are distributed as $\mathcal{N}(t_n|f(x_n), \sigma^2)$, where σ^2 is the (known) variance of the noise. We shall now try to describe the data with a model $y(x, \boldsymbol{w})$, where $\boldsymbol{w}$ is a vector of parameters determining the model. It could for example be the coefficients in the polynomial model studied in the introduction (Eq. 2.1), but at this point we do not restrict the way that the model depends on the parameters.
We are interested in determining the posterior probability distribution of the parameters and we therefore use Bayes' theorem (what else!):

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) = \prod_{n=1}^{N} \left(\frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(t_n - y(x_n, \boldsymbol{w}))^2/2\sigma^2} \right) \mathcal{P}(\boldsymbol{w}) \quad (11.1)$$

$$= \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-\sum_n (t_n - y(x_n, \boldsymbol{w}))^2/2\sigma^2} \mathcal{P}(\boldsymbol{w}) \quad (11.2)$$

If we for a moment neglect the prior, we see that the highest probability is obtained by maximizing the likelihood, which again amounts to minimizing the cost function $C = \sum_n (t_n - y(x_n, \boldsymbol{w}))^2$. This is exactly the least-squares method that we used for the polynomial fits in the introduction. However, what we discovered there, was that this approach could lead to severe overfitting, and this is where the prior comes into play.
Why do we think that the nineth order polynomial in Fig. 2.1 (right) is crazy? It does not do well on the sine function overall, but it does go through all the data points so in that sense it is a legitimate suggestion for the function given the data. In fact, if we from the outset hadn't picked the sine function, but this nineth order polynomial itself as our target function, the fit would have been great! This is where we have to look at the prior. The prior gives the opportunity to describe expectations that we might have to the function we are trying to fit, even before seeing any data. In the case of the sine function, we might for example expect that the function values lie between, say, -2 and 2, and that overall the function is fairly smooth and only varies on the length scale of 1.

How can we put this information into the prior? If we look at the polynomial expression as the first part of a Taylor expansion, the coefficients correspond to derivatives $f^{(i)}(x = 0) = i!\, w_i$. If we expect that the derivatives should not be too large, this automatically sets an expectation for the coefficients w_i. Would such a restriction do any good? Well, if we look at the coefficients in the nineth order polynomial above, several of these are larger than 10^5, and the eighth derivative ends up being of the order $2 \cdot 10^{10}!$ So applying reasonable restrictions to the parameters, would definitely get rid of this crazy solution. We shall take up different ways of implementing the prior below, but first we shall discuss the broadly used class of models, which are linear in the model parameters.

Chapter 12

Linear basis function models

A particularly simple kind of models are those, which are linear in the model parameters. These models take the form

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \tag{12.1}$$

where $\phi_i(\boldsymbol{x})$ is a set of basis functions used for the expansion of the model function. $\boldsymbol{\phi}$ denotes the (column) vector of the functions $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_M)$. The example in the introduction corresponds to the power functions $\phi_i(x) = x^{i-1}$.
Assuming Gaussian noise on the data we get the same posterior probability distribution as in the previous chapter:

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) = \mathcal{P}(\boldsymbol{t}|\boldsymbol{w})\mathcal{P}(\boldsymbol{w}) = \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-\sum_n (t_n - y(x_n, \boldsymbol{w}))^2 / 2\sigma^2} \mathcal{P}(\boldsymbol{w}) \tag{12.2}$$

$$= \frac{1}{(2\pi\sigma^2)^{(N/2)}} e^{-C(\boldsymbol{w})/\sigma^2} \mathcal{P}(\boldsymbol{w}), \tag{12.3}$$

where the cost function is given by

$$C(\boldsymbol{w}) = \frac{1}{2} \sum_n (t_n - \sum_i w_i \phi_i(\boldsymbol{x}_n))^2 = \frac{1}{2} (\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})^2. \tag{12.4}$$

Here we have introduced the so-called *design matrix* $\boldsymbol{\Phi}$, which is an $N \times M$ matrix with components $\boldsymbol{\Phi}_{nj} = \phi_j(x_n)$. (Note that the indices are switched!)

Chapter 13

Least squares method revisited

In the least squares method we optimize the likelihood, which corresponds to minimizing the cost function. This is easily done by calculating the gradient of the cost function with respect to the model parameters and setting it to zero

$$\frac{\partial C}{\partial \boldsymbol{w}} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w} - \boldsymbol{\Phi}^T\boldsymbol{t} = 0, \tag{13.1}$$

with the solution

$$\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}. \tag{13.2}$$

The quantity $(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T$ is called the *Moore-Penrose inverse* matrix. If $\boldsymbol{\Phi}$ was an invertible matrix, it would be the usual inverse matrix, however, $\boldsymbol{\Phi}$ is usually not invertible. It is an $N \times M$ matrix and therefore in general not even quadratic. The matrix $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ is an $M \times M$ matrix and thus corresponds to a linear mapping in parameter space. $\boldsymbol{\Phi}$ maps from parameter space to data space and $\boldsymbol{\Phi}^T$ maps back. The product is therefore only invertible if $M \leq N$, *i.e.* if we have more data points than parameters. In the opposite case the data are not sufficient to determine the parameters, and $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ becomes singular. (A linear function that maps to a vector space of smaller dimension and back, cannot be invertible).

Exercise: Method of least squares

Write some Python code in a Jupyter notebook to solve the least-squares fitting problem and produce figures similar to the ones in the introduction. You need to

1. Define the function $f(x) = \sin(2\pi x)$ that we want to fit.
2. Now we generate some data points with noise. We shall take the noise to be Gaussianly distributed with variance $\sigma^2 = 0.1^2$. this can be done in the following way:

```
N = 10
np.random.seed(seed=54)
xp = np.sort(np.random.random_sample(N))
np.random.seed(seed=58)
sigma = 0.1
tp = f(xp) + np.random.normal(0,sigma,N)
```

where f is defined as $f(x) = \sin(2\pi x)$. This should hopefully give you the following data:

xp	tp
0.00860545	-0.0219753
0.029657	-0.0248944
0.184877	0.836471
0.363239	0.756628
0.420183	0.31031
0.518283	-0.0564809
0.671484	-0.977676
0.757312	-1.14703
0.801381	-1.00067
0.968936	-0.0672362

This is the data used for the examples in the introduction. If you stick to these data, you shall also be able to compare results.

3. Define your basis functions for the fit (*i.e.* the polynomial functions).
4. Calculate the design matrix $\mathbf{\Phi}$.
5. Find the expansion parameters from Eq. 13.2.
6. Plot your fit (Eq. 12.1) to the sine function.
7. We are so lucky that we know the full function we are trying to fit, which is of course usually not the situation. We can therefore estimate the quality of the fit by calculating the error $\int (f(x) - y(x, \boldsymbol{w}_0))^2 \, dx$.
8. Play with the different parameters: the number of data points; the amount of noise; the number of model parameters (*i.e.* the order of the polynomial) etc.
9. When do you observe overfitting? Can you see that the parameters become very large in those cases?

Chapter 14

Regularized least squares method – including the prior

We have seen that the least-squares approach, where we just maximize the likelihood, may lead to significant overfitting, and we have also briefly discussed how to fix it using the prior. We shall now go into that in more detail.
A simple and popular choice for the prior probability distribution of the parameters is a Gaussian (of course, what else?)

$$\mathcal{P}(\boldsymbol{w}) \propto e^{-\frac{1}{2}\lambda \boldsymbol{w}^T \boldsymbol{w}} = e^{-\frac{1}{2}\lambda \sum_i |w_i|^2}, \tag{14.1}$$

where λ is a constant. This prior expresses the expectation that all parameters are of a reasonable size and suppresses the probability for large values of w_i. How much we want to punish large parameter values is controlled by the value of λ.
Including the prior in the description, we will now maximize not only the likelihood but the posterior probability (likelihood times prior). This corresponds to minimizing a revised cost function

$$C_{\text{reg}}(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{t} - \boldsymbol{\Phi}\boldsymbol{w})^2 + \frac{1}{2}\tilde{\lambda}\boldsymbol{w}^T\boldsymbol{w}, \tag{14.2}$$

with $\tilde{\lambda} = \sigma^2\lambda$. The last term on the right hand side is also called a *regularization term*, and the cost function is said to be *regularized*. Since the regularization term is also quadratic in the parameters, it is quite straightforward to perform the minimization, which leads to the optimal model parameters

$$\boldsymbol{w}_0 = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}\boldsymbol{\Phi}^T\boldsymbol{t}. \tag{14.3}$$

Prediction for new values of the variable $\boldsymbol{x}$ can then be made as

$$y(\boldsymbol{x}, \boldsymbol{w}_0) = \boldsymbol{w}_0^T\boldsymbol{\phi}(\boldsymbol{x}) \tag{14.4}$$

A couple of remarks is at its place here:

- We have derived the regularization term in the cost function from Bayesian analysis. However, regularization terms are often in machine learning introduced

rather pragmatically as a means to control model complexity and avoid overfitting. The particular quadratic form, which here is derived based on the assumption of a Gaussian prior, is convenient because the minimum of the cost function can be obtained by a single matrix inversion (Eq. 14.3), but other choices are also possible. The quadratic regularization is also called L_2-regularization because of the corresponding norm $||\boldsymbol{w}||_2^2 = \sum_i w_i^2$. Another choice is the L_1-norm, which gives a regularization term of the form $\tilde{\lambda}||\boldsymbol{w}||_1 = \tilde{\lambda}\sum_i |w_i|$ in the cost function. This method is called LASSO ("least absolute shrinkage and selection operator") and it corresponds to a prior probability given by the Laplace distribution. The LASSO is more complicated to handle because the minimum cannot be obtained by a single matrix inversion, but the method has the virtue that it suppresses some parameters to be not only small, but exactly zero. This results in simpler models with fewer parameters.

- The value of the regularization parameter λ is in the Bayesian setting given as part of the prior probability distribution. Such a parameter is also called a *hyperparameter*, and we shall later discuss how to determine this using Bayes' theorem. However, an alternative and common way to treat such parameters in machine learning is through *cross validation*, which goes something like:

  The data set is divided up in three parts the *training* set, the *validation* set, and the *test* set. The model is constructed using the training set and then tried on the validation test. The hyperparameter $\tilde{\lambda}$ is then changed so that the train/validation procedure leads to the optimal performance on the the validation set. If the value of $\tilde{\lambda}$ is too small, the model is too flexible leading to large variance, while if $\tilde{\lambda}$ is too large, the model is too simple resulting in high bias. Finally, after the determination of the optimal hyperparameter, the model is applied to the test set in order to get an idea of the performance of the model on new data.

Exercise: Regularized least squares method

This is a continuation of the previous exercise on the method of least squares, but now with regularization. The exercise proceeds in the same way, but the optimal values for the model parameters should now be determined from Eq. 14.3. The calculated error $\int (f(x) - y(x, \boldsymbol{w}_0))^2\, dx$ therefore becomes a function of the regularization parameter $\tilde{\lambda}$. (When writing Python code please note that "lambda" is a reserved name, so you have to call your variable something else). Because we now use regularization we can make the model more complex and we therefore consider polynomials up to degree 19 (*i.e.*, with 20 parameters).

1. Try different values of $\tilde{\lambda}$ and plot your fit (Eq. 12.1) to the sine function.
2. Plot the error of the fit ($\int (f(x) - y(x, \boldsymbol{w}_0))^2\, dx$) as a function of the regularization parameter $\tilde{\lambda}$. Consider $\tilde{\lambda}$ on a logarithmic scale. What is a good value for $\tilde{\lambda}$?
3. Play with the different parameters: the number of data points; the amount of noise; the number of terms in the polynomial expansion.
4. If you have time you can also try the L_1-regularization. In that case you have to use a numerical method to minimize the cost as a function of the

parameters. Observe that now some parameters become exactly zero and can therefore be removed from the model.

Chapter 15

Beyond the least squares model – the posterior distribution

In the previous chapters, we first maximized the likelihood in the simplest least square fit, and then we included the prior and maximized the posterior. But hey, this is not fully Bayesian! Why should we be satisfied with only the maximum point for the posterior, when the whole distribution is available? This means that we can get not only a "best fit" function but a complete distribution of functions including an estimate for the uncertainty of the predictions.
Because of the Gaussian nature of both the prior and the likelihood, the cost function is quadratic in the parameters. Using the expression of the cost function Eq. 14.2 and the values for the optimal parameters Eq. 14.3, it is straightforward to show that the cost function can be written

$$C(\boldsymbol{w}) = C_0 + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)(\boldsymbol{w} - \boldsymbol{w}_0), \tag{15.1}$$

where C_0 is a constant. In other words the posterior distribution becomes

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{w} - \boldsymbol{w}_0)^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)(\boldsymbol{w} - \boldsymbol{w}_0)\right) \tag{15.2}$$

$$= \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^T\boldsymbol{\Omega}^{-1}(\boldsymbol{w} - \boldsymbol{w}_0)\right) \tag{15.3}$$

$$= \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Omega}), \tag{15.4}$$

where the matrix $\boldsymbol{\Omega}$ is defined by

$$\boldsymbol{\Omega} = \sigma^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \tilde{\lambda}\boldsymbol{I}_M)^{-1}, \tag{15.5}$$

and where $\mathcal{N}$ denotes a multi-dimensional Gaussian distribution. You can read more about this distribution in the appendices, but the main point is that the average and the covariance matrix are given by

$$\langle \boldsymbol{w} \rangle = \boldsymbol{w}_0 \text{ and } \text{Cov}[\boldsymbol{w}, \boldsymbol{w}^T] = \boldsymbol{\Omega}. \tag{15.6}$$

Having the posterior distribution of the parameters under control, we can move on to the distribution of the predicted values. Since the fitted functions are linear in the

parameters (Eq. 14.4), we can write

$$y(x, \boldsymbol{w}) = y_0(x) + (\boldsymbol{w} - \boldsymbol{w}_0)^T \boldsymbol{\phi}(x), \tag{15.7}$$

where y_0 is the average prediction corresponding to the parameters $\boldsymbol{w}_0$ (Eq. 14.4). Finally, the covariance of the predictions at two different points x and x' are

$$\begin{aligned} \text{Cov}[y(x, \boldsymbol{w}), y(x', \boldsymbol{w})] &= \langle (y(x, \boldsymbol{w}) - y_0(x))(y(x', \boldsymbol{w}) - y_0(x')) \rangle & (15.8) \\ &= \boldsymbol{\phi}^T(x) \langle (\boldsymbol{w} - \boldsymbol{w}_0)(\boldsymbol{w} - \boldsymbol{w}_0)^T \rangle \boldsymbol{\phi}(x') & (15.9) \\ &= \boldsymbol{\phi}^T(x) \boldsymbol{\Omega} \boldsymbol{\phi}(x'). & (15.10) \end{aligned}$$

In particular, if $x = x'$ we have

$$\text{Cov}[y(x, \boldsymbol{w}), y(x, \boldsymbol{w})] = \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle = \boldsymbol{\phi}^T(x) \boldsymbol{\Omega} \boldsymbol{\phi}(x). \tag{15.11}$$

We now have all the ingredients to make a proper Bayesian analysis of the fitting problem. And that is what we are going to do in the next exercise.

Exercise: Bayesian analysis of a linear basis function model

In this exercise we shall improve on the least square analysis in the two previous exercises. In the above exercise you should have constructed some data points from a sine function with some noise σ. Furthermore, you have found a good value for the regularization parameter $\tilde{\lambda}$, which has a good compromise between bias and variance. We shall use $\tilde{\lambda} = 10^{-4}$ in the following. Knowing these parameters, you should be able to construct the matrix $\boldsymbol{\Omega}$. Now, try the following

- First look at the prior distribution of polynomial functions, *i.e.* the distribution of functions that we have before we have even considered the data. This is given by Eq. 14.1. Generate from this distribution an ensemble of parameters and plot the corresponding functions. Do these functions look like a good starting point? What happens if you change the parameter $\tilde{\lambda}$?
- And now we move on to the posterior distribution. Generate an ensemble of vectors from the multi-dimensional Gaussian distribution Eq. 15.4. This can be done using the Python numpy function random.multivariate_normal. Each of these vectors gives rise to a predicted function through Eq. 14.4. Plot all the predicted functions in one plot together with the sine function, the data points, and the optimal model.
- The expression Eq. 15.11 determines the width of the predictions or in other words the error bar. Plot the average predicted function (the one from $\boldsymbol{w}_0$) and a shaded area around the function indicating the error bar.

Chapter 16

Kernel regression

We shall now approach the fitting problem in an apparently different way, namely through interpolation. But as we shall see later this alternative approach is in fact very closely related to the linear models we already investigated.
We shall again assume that we know the function $f(x)$ at a number of points $y_n = f(x_n),\ n = 1, 2, \ldots, N$, and we want to predict the value $f(x)$ at a new point x. If the point x is very close to one of the data points, say x_m, we might expect, that the function value will also be close $f(x) \sim f(x_m)$. We therefore define a function $g(x, x')$, which "measures" if two points x and x' are close. The function has the properties $g(x, x) = 1$ and $g(x, x') \to 0$ if $|x - x'| \to \infty$. We can for example use the function

$$g(x, x') = \exp(-(x - x')^2/2\ell^2), \tag{16.1}$$

where ℓ is a parameter to be discussed further later. Please note, that we here use a Gaussian function, but it does not play the role of a probability distribution. We could for example also have used the function $\exp(-|x - x\,\prime|/\ell)$. The function g is closely related to the so-called *kernel function*, we shall define in the next chapter.
We now approximate the function by a sum over the data points using the function g

$$y(x, \boldsymbol{a}) = \sum_{n=1}^{N} g(x, x_n) a_n = \boldsymbol{g}(x)^T \boldsymbol{a}, \tag{16.2}$$

where $\boldsymbol{g}(x)^T = (g(x, x_1), g(x, x_2), \ldots, g(x, x_N))$. This formula can be compared to the linear expansion over basis functions in Eq. 12.1, but here we have N terms in the sum corresponding to the data points instead of M terms corresponding to the basis functions.
If we neglect noise we would ideally like the model to reproduce all the data points (x_n, y_n), and we now realize, that we have exactly the right number of parameters $a_1, a_2, \ldots, a_N$ to do this! Introducing the matrix $\boldsymbol{G}_{nm} = g(x_n, x_m)$ the condition that the model reproduces the data points is $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{a}_0$ with the solution $\boldsymbol{a}_0 = \boldsymbol{G}^{-1}\boldsymbol{y}$. Prediction of a new point thus becomes

$$y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{a}_0 = \boldsymbol{g}(x)^T \boldsymbol{G}^{-1} \boldsymbol{y}. \tag{16.3}$$

Exercise: Kernel regression

We now return to our great attempts to fit a sine function. Use Eq. 16.3 to predict the function in the case where there is no noise in the data. Try different values of the parameter ℓ. What happens if you try to use the formula in the case with noise?

What should we do if there is noise on the data? Using the formula Eq. 16.3 on the noisy data $y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{G}^{-1} \boldsymbol{t}$ will at best reproduce the noisy data points leading to severe overfitting. At worst it might even be impossible to invert the matrix $\boldsymbol{G}$ because it is (close to being) singular. The solution is to apply the least-squares approach with regularization. We thus define a cost function

$$C_{\text{reg}} = (\boldsymbol{t} - \boldsymbol{G}\boldsymbol{a})^2 + \lambda' \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a}, \tag{16.4}$$

where the second term is preventing unreasonable large parameters $\boldsymbol{a}$. The introduction of the matrix $\boldsymbol{G}$ in the regularization ensures that the regularization is "local" because $\boldsymbol{G}_{nm} = 0$ if the points x_n and x_m are far from each other.
The minimization of the cost function leads to the solution $\boldsymbol{a}_0 = (\boldsymbol{G} + \lambda' \boldsymbol{I}_M)^{-1} \boldsymbol{t}$, and the predicted function is therefore given by

$$y(x, \boldsymbol{a}_0) = \boldsymbol{g}(x)^T \boldsymbol{a}_0 = \boldsymbol{g}(x)^T (\boldsymbol{G} + \lambda' \boldsymbol{I}_N)^{-1} \boldsymbol{t}. \tag{16.5}$$

The regularization parameter λ' makes the matrix inversion in Eq. 16.5 well-behaved.

Exercise: Kernel ridge regression

The kernel regression with a regularization is sometimes called kernel ridge regression. Apply the method to our beloved sine problem. Vary the regularization parameter λ' and see how it affects the fit and the calculated error $\int (f(x) - y(x, \boldsymbol{a}_0))^2 \, dx$.

Chapter 17

Gaussian Processes

Now the time has come to see the linear models and the kernel regression in a new and revealing light. We shall see that we can get entirely rid of the basis functions since everything can be expressed through the so-called *kernel function*

$$k(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{\lambda} \sum_{i=1}^{M} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}'). \tag{17.1}$$

At the same time we shall shift from using the coefficients $\boldsymbol{w}$ as the basic variables to directly work with the predicted function $y(\boldsymbol{x})$. This approach is also called a *Gaussian process*.

Let us first look at the prior probability distribution, which was given in terms of the parameters $\boldsymbol{w}$ (see Eq. 14.1) as a Gaussian distribution $\mathcal{P}(\boldsymbol{w}) \propto \exp(-\lambda \boldsymbol{w}^T \boldsymbol{w}/2)$. Since the function is a linear combination of the parameters $y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}$, the function itself (for any point $\boldsymbol{x}$) will also have a Gaussian prior distribution. (The Gaussian distribution is an exponential with a second order polynomial in the exponent. If we make a linear transformation in the exponent, we get a second order polynomial in the new variables, *i.e.* the result is a new Gaussian distribution). We can find the Gaussian distribution for the function by calculating the average and the covariance. The average is easy and given by

$$\langle y(\boldsymbol{x}) \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w} \rangle = 0, \tag{17.2}$$

since the parameters are distributed around 0.

The covariance is given by

$$\text{Cov}[y(\boldsymbol{x}), y(\boldsymbol{x}')] = \langle y(\boldsymbol{x}) y(\boldsymbol{x}') \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w}\boldsymbol{w}^T \rangle \boldsymbol{\phi}(\boldsymbol{x}') \tag{17.3}$$

$$= \frac{1}{\lambda} \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{I}_M \boldsymbol{\phi}(\boldsymbol{x}') = \frac{1}{\lambda} \sum_{i=1}^{M} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}'). \tag{17.4}$$

We can thus formally write the prior distribution

$$\mathcal{P}([y(\boldsymbol{x})]) = \mathcal{N}([y(\boldsymbol{x})] | 0, k(\boldsymbol{x}, \boldsymbol{x}')) \tag{17.5}$$

We use the square brackets to indicate that it is the whole function, which is the variable.

Eq. 17.5 means that we have a direct interpretation of the kernel function: it is the function that determines our prior distribution of functions! As an example of a kernel function for our one-dimensional sine problem we can use the Gaussian form $k(x, x') = \exp(-(x - x')^2/2\ell^2)$ (yes, this looks very much like our function g above, it will soon be clear why). This kernel function cannot be written in the form of a finite number of basis functions as in Eq. 17.1. We can regard it as a limit with infinitely many basis functions, but it really does not matter. We can just take the kernel function directly as the starting point. The interpretation of the parameter ℓ is now fairly clear from Eq. 17.4: if $|x - x'| \gg \ell$ the function values $y(x)$ and $y(x')$ are uncorrelated, however, if $|x - x'| \leq \ell$ the function values $y(x)$ and $y(x')$ will tend to be close to each other.

Exercise: Prior distribution with Gaussian kernel function

Generate in the one-dimensional case an ensemble of functions pulled from the prior distribution Eq. 17.5 with the Gaussian kernel. Since the variable x is continuous we have to approximate it on a grid of points $x_p = p/N_g$, with $p = 1, 2, \ldots, N_g$. You have already done this for all the functions you have plotted. The kernel function $k(x, x')$ thus becomes a (large) matrix at the points $k(x_p, x_q)$.

Try to vary the parameter ℓ. How do the generated functions change? Compare to the prior distribution of functions when using the polynomials.

We are now ready to consider the posterior distribution. Again referring to the linear model with basis functions, the posterior distribution is Gaussian in the model parameters, and since the function $y(\boldsymbol{x})$ is linear in the parameters, the distribution of the function also becomes Gaussian. We therefore only need to determine the average value and the covariance.
The average value of the predicted function is by using Eq. 14.3 given by

$$y_0(\boldsymbol{x}) := \langle y(\boldsymbol{x}) \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \langle \boldsymbol{w} \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}_0 \tag{17.6}$$

$$= \boldsymbol{\phi}(\boldsymbol{x})^T (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \tilde{\lambda} \boldsymbol{I}_M)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t}. \tag{17.7}$$

We now use the sneaky formula

$$(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \tilde{\lambda} \boldsymbol{I}_M)^{-1} \boldsymbol{\Phi}^T = \boldsymbol{\Phi}^T (\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \tilde{\lambda} I_N)^{-1}, \tag{17.8}$$

which can be shown by multiplying with the two inverse matrices. We also define the vector function $\boldsymbol{k}(\boldsymbol{x})$ with components ($n = 1, 2, \ldots, N$)

$$k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n) = \left(\frac{1}{\lambda} \boldsymbol{\Phi} \boldsymbol{\phi}(\boldsymbol{x}) \right)_n, \tag{17.9}$$

and the $N \times N$ *kernel matrix*, $\boldsymbol{K}$, with components

$$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \frac{1}{\lambda} \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^T \right)_{nm}. \tag{17.10}$$

This allows us to rewrite

$$y_0(\boldsymbol{x}) = \langle y(\boldsymbol{x}) \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Phi}^T (\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \tilde{\lambda} I_N)^{-1} \boldsymbol{t} \tag{17.11}$$

$$= \boldsymbol{k}(\boldsymbol{x})^T (\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)^{-1} \boldsymbol{t} = \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{t}, \tag{17.12}$$

where we furthermore introduced the matrix $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$.
And voila, everything is written in terms of the kernel without reference to the individual basis functions!
We furthermore realize that the prediction of the average function is exactly the same as obtained in the previous chapter if we identify the function g with the kernel function k and λ' with the noise! This provides an easy way to understand the average prediction of the Gaussian process: we just "drop" a function $k(\boldsymbol{x}, \boldsymbol{x}_n)$ at each data point $\boldsymbol{x}_n$ and make a linear combination of the functions to fit the data points – including regularization.
What about the error bar on the prediction? We know from the linear models that (see Eq. 15.11)

$$\sigma^2(\boldsymbol{x}) := \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle = \boldsymbol{\phi}^T(x) \sigma^2 \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \tilde{\lambda} \boldsymbol{I}_M \right)^{-1} \boldsymbol{\phi}(x). \tag{17.13}$$

We now introduce another sneaky matrix formula

$$\left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \tilde{\lambda} \boldsymbol{I}_M \right)^{-1} = \frac{1}{\tilde{\lambda}} \left(1 - \boldsymbol{\Phi}^T \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^T + \tilde{\lambda} \boldsymbol{I}_N \right)^{-1} \boldsymbol{\Phi} \right), \tag{17.14}$$

which can be shown by multiplying by the inverse on the left hand side. Using the formula we get the simple final result

$$\sigma^2(\boldsymbol{x}) = \langle (y(x, \boldsymbol{w}) - y_0(x))^2 \rangle = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{k}(\boldsymbol{x}). \tag{17.15}$$

Together with the result for the average this defines the posterior probability distribution obtained with the Gaussian process.
The prior functions are Gaussianly distributed around the function zero. However, we can change the prior average function to some other function $y^{(p)}(\boldsymbol{x})$ by subtracting this from all functions and adding it to the result. Including this feature the Gaussian process can be summarized as follows:

Gaussian process. Given the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ we can define the vector function $\boldsymbol{k}(\boldsymbol{x})$ by $k_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n)$ and the kernel matrix $\boldsymbol{K}$ by $K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_n)$. We furthermore define the matrix $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$. The prior average function is denoted $y_p(\boldsymbol{x})$ with the corresponding vector $(\boldsymbol{y}_p)_n = y_p(\boldsymbol{x}_n)$. With these definitions we get the prediction for the average function $y_0(\boldsymbol{x})$ and the variance $\sigma^2(\boldsymbol{x})$ as

$$y_0(\boldsymbol{x}) = y_p(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} (\boldsymbol{t} - \boldsymbol{y}_p) \tag{17.16}$$

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{k}(\boldsymbol{x}) \tag{17.17}$$

Some remarks on this result:

Connection to linear basis function models. We derived the formulas Eq. (17.16) and 17.17 for the Gaussian process using a kernel function expressed in terms of linear basis function. The results therefore necessarily agree with the ones obtained with linear basis function models. However, the Gaussian process is more general because the kernel function does not have to be expressed in terms of a

finite number of basis functions. The Gaussian kernel function can for example only be written using an infinite number of basis functions.

Working with a kernel is often more intuitive than working with a set of basis functions. We saw for example above that using the polynomials resulted in a rather different description of the region around $x = 0$ compared to the region around $x = 1$, which was maybe hard to foresee. With, for example, the Gaussian kernel, it is clear that all regions will be treated the same way since the function depends on only the distance $|x - x'|$. Furthermore, the length scale ℓ can be directly interpreted as the length scale of the functions in the prior probability distribution.

Points in unexplored regions. If we consider a point, $\boldsymbol{x}$, far away from all data points, $\boldsymbol{x}_n$, then $\boldsymbol{k}(\boldsymbol{x}) = 0$, and the prediction for the average becomes $y_0(\boldsymbol{x}) = y_p(\boldsymbol{x})$ and for the variance it reduces to $\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x})$. This should not be a surprise, since this is exactly the same properties we have in the *prior* distribution: The distribution is around the function $y_p(\boldsymbol{x})$ with a variance of $k(\boldsymbol{x}, \boldsymbol{x})$. So if a point is far away from all previous data points (on the scale determined by the kernel function) we just get back our prior expectation. This also shows that the value $k(\boldsymbol{x}, \boldsymbol{x})$ is important for the prediction of the variance. The Gaussian kernel function is therefore properly defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_0 \exp(-(\boldsymbol{x} - \boldsymbol{x}')^2/2\ell^2) \tag{17.18}$$

with a constant k_0 in front. This constant was set to one in the exercise with the kernel ridge regression, and this is ok to do, if we look at only the average predicted function $y_0(\boldsymbol{x})$. In that case the constant appears in both $\boldsymbol{k}(\boldsymbol{x})$ and $\boldsymbol{C}$ and therefore cancels out in the prediction (with a suitable rescaling of the regularization parameter λ'). However, the prefactor is crucial for the prediction of the variance Eq. 17.17. So the Gaussian kernel contains two hyperparameters k_0 and ℓ.

Exercise: Gaussian process for the sine function

We are now ready to analyze the fitting of the sine function using a Gaussian process. Calculate the average predicted function and the variance using the previously constructed noisy data points. Plot the average function with a shaded area around it representing the (square root of) the variance. Try different values of the hyperparameters k_0 and ℓ and see what happens.

Try also to use the Gaussian process on the data points without noise, setting $\sigma = 0$ so that $\boldsymbol{C} = \boldsymbol{K}$.

Chapter 18

Derivation of the Gaussian process with functionals (optional)

In this chapter we derive the Gaussian process using functionals. It is an elegant description, which quickly gets to the basic equations for the Gaussian process, but it is more advanced mathematically.

Functions $y(\boldsymbol{x})$ can be regarded as vectors, where the variable $\boldsymbol{x}$ corresponds to the index for the vector, but is continuous. We shall write this vector as $\underline{y}$. The dot product of two functions $y(\boldsymbol{x})$ and $y'(\boldsymbol{x})$ is accordingly given by $\underline{y}^T\underline{y}' = \int y(\boldsymbol{x})y'(\boldsymbol{x})\,d\boldsymbol{x}$, where the usual summation over an index is substituted by an integration over $\boldsymbol{x}$. An operator $\hat{g}$ is described by a "matrix" $g(\boldsymbol{x},\boldsymbol{x}')$ with two "indices" $\boldsymbol{x}$ and $\boldsymbol{x}'$. An operator is applied to a function as $\hat{g}\,\underline{y} = \int g(\boldsymbol{x},\boldsymbol{x}')y(\boldsymbol{x}')\,d\boldsymbol{x}'$. The identity operator thus becomes $\hat{1} = \delta(x - x')$.

And now for the derivation of the Gaussian process. We start from Bayes' theorem in the form

$$\mathcal{P}(\underline{y}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\underline{y})\mathcal{P}(\underline{y}). \tag{18.1}$$

The probabilities depend on the whole function $y(\boldsymbol{x})$, *i.e.* they are *functionals*. The data are the values $t_n, n = 1, 2, \ldots, N$ corresponding to the input values $\boldsymbol{x}_n$.

We start with the prior, where we shall assume a Gaussian distribution of $y(\boldsymbol{x})$ with the average and covariance given by

$$\langle y(\boldsymbol{x})\rangle = y_p(\boldsymbol{x}) \;\text{ and }\; \mathrm{Cov}[y(\boldsymbol{x})y(\boldsymbol{x}')] = k(\boldsymbol{x},\boldsymbol{x}'). \tag{18.2}$$

Here, $y_p(\boldsymbol{x})$ is our prior average expectation for the function, and the kernel $k(\boldsymbol{x},\boldsymbol{x}')$ is introduced directly as the covariance. This can be written in the short vector notation as $\langle\underline{y}\rangle = \underline{y_p}$ and $\mathrm{Cov}[\underline{y}\underline{y}^T] = \hat{k}$.

We know that for the multi-dimensional Gaussian distribution the inverse of the covariance appears in the exponent ((Chapter 21.1.1)), so the prior distribution is given by

$$\mathcal{P}(\underline{y}) = \mathcal{N}(\underline{y}|\underline{y_p},\hat{k}) \tag{18.3}$$

$$\propto \exp\left(-\frac{1}{2}\int (y(\boldsymbol{x}) - y_p(\boldsymbol{x}))k^{-1}(\boldsymbol{x},\boldsymbol{x}')(y(\boldsymbol{x}') - y_p(\boldsymbol{x}))\,d\boldsymbol{x}\,d\boldsymbol{x}'\right) \tag{18.4}$$

$$= \exp\left(-\frac{1}{2}(\underline{y}^T - \underline{y_p}^T)\hat{k}^{-1}(\underline{y} - \underline{y_p})\right) \tag{18.5}$$

where the inverse in the exponent is defined through $\hat{k}^{-1}\hat{k} = \hat{1}$ or, written out in coordinates, $\int k^{-1}(\boldsymbol{x}, \boldsymbol{x}'')k(\boldsymbol{x}'', \boldsymbol{x}')\, d\boldsymbol{x}'' = \delta(\boldsymbol{x} - \boldsymbol{x}')$.
The likelihood is given the usual way assuming Gaussian noise with variance σ^2:

$$\mathcal{P}(\boldsymbol{t}|\underline{y}) = \mathcal{N}(\boldsymbol{t}|\underline{y}, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y(\boldsymbol{x}_n) - t_n)^2\right) \tag{18.6}$$

$$= \exp\left(-\frac{1}{2}\int (y(\boldsymbol{x}) - t(\boldsymbol{x}))q(\boldsymbol{x}, \boldsymbol{x}')(y(\boldsymbol{x}') - t(\boldsymbol{x}'))\, d\boldsymbol{x}\, d\boldsymbol{x}'\right) \tag{18.7}$$

$$= \exp\left(-\frac{1}{2}(\underline{y}^T - \underline{t}^T)\hat{q}(\underline{y} - \underline{t})\right), \tag{18.8}$$

where we have introduced the operator $\hat{q}$, which is expressed in coordinates as

$$q(\boldsymbol{x}, \boldsymbol{x}') = \sum_{n=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}_n)\frac{1}{\sigma^2}\delta(\boldsymbol{x}' - \boldsymbol{x}_n). \tag{18.9}$$

$t(\boldsymbol{x})$ is here any function, which fulfills $t(\boldsymbol{x}_n) = t_n$, since the operator $\hat{q}$ only picks out values at $\boldsymbol{x}_n$.
The posterior, which is the product of the prior and the likelihood, can now be written

$$\mathcal{P}(\underline{y}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2}(\underline{y}^T - \underline{t}^T)\hat{q}(\underline{y} - \underline{t}) - \frac{1}{2}(\underline{y}^T - \underline{y_p}^T)\hat{k}^{-1}(\underline{y} - \underline{y_p})\right). \tag{18.10}$$

Since the exponent is quadratic in y, the posterior can also be written as

$$\mathcal{P}(\underline{y}|\boldsymbol{t}) \propto \exp\left(-\frac{1}{2}(\underline{y}^T - \underline{y_0}^T)\hat{g}^{-1}(\underline{y} - \underline{y_0})\right), \tag{18.11}$$

where it is easy to see that the "curvature" of the quadratic function is given by

$$\hat{g}^{-1} = \hat{k}^{-1} + \hat{q}. \tag{18.12}$$

The function, $y_0(\boldsymbol{x})$, that maximizes the posterior distribution can be found by taking the derivative of the exponent with respect to $\underline{y}^T$ and setting it to zero. The result is

$$\underline{y_0} = \underline{y_p} + \hat{g}\hat{q}(\underline{t} - \underline{y_p}), \tag{18.13}$$

or written out in coordinates using Eq. 18.9

$$y_0(\boldsymbol{x}) = y_p(\boldsymbol{x}) + \frac{1}{\sigma^2}\sum_{n=1}^{N} g(\boldsymbol{x}, \boldsymbol{x}_n)(t_n - y_p(\boldsymbol{x}_n)). \tag{18.14}$$

At this point we have in principle solved the problem: $y_0(\boldsymbol{x})$ is the average predicted function and the covariance is $\mathrm{Cov}[y(\boldsymbol{x})y(\boldsymbol{x}')] = g(\boldsymbol{x}, \boldsymbol{x}')$, since the inverse of g appears in the exponent in Eq. 18.11.
But how do we evaluate $g(\boldsymbol{x}, \boldsymbol{x}')$? To this end we consider the operator

$$\hat{d} = \left(\hat{1} + \hat{q}\hat{k}\right)^{-1}\hat{q}. \tag{18.15}$$

By multiplying with $\hat{1} + \hat{q}\hat{k}$ from the left on both sides of the equation, we see that the operator d in coordinates can be written as

$$d(\boldsymbol{x}, \boldsymbol{x}') = \sum_{nm} \delta(\boldsymbol{x} - \boldsymbol{x}_n) \left(\boldsymbol{C}^{-1}\right)_{nm} \delta(\boldsymbol{x}' - \boldsymbol{x}_m), \tag{18.16}$$

where the matrix $\boldsymbol{C}$ has the components

$$\boldsymbol{C}_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \sigma^2 \delta_{nm}. \tag{18.17}$$

Using this we can now write the average predicted function as

$$\underline{y_0} = \underline{y_p} + \hat{g}\hat{q}(\underline{t} - \underline{y_p}) = \underline{y_p} + (\hat{k}^{-1} + \hat{q})^{-1}\hat{q}(\underline{t} - \underline{y_p}) \tag{18.18}$$

$$= \underline{y_p} + \hat{k}(\hat{1} + \hat{q}\hat{k})^{-1}(\underline{t} - \underline{y_p}) = \underline{y_p} + \hat{k}\hat{d}(\underline{t} - \underline{y_p}), \tag{18.19}$$

or using coordinates

$$y_0(\boldsymbol{x}) = y_p(\boldsymbol{x}) + \sum_{nm} k(\boldsymbol{x}, \boldsymbol{x}_n)(\boldsymbol{C}^{-1})_{nm}(t_n - y_p(\boldsymbol{x}_n)), \tag{18.20}$$

which is the final result for the average function.
We still need the covariance, which we saw is given by the function g. We rewrite

$$\hat{g} = (\hat{k}^{-1} + \hat{q})^{-1} = \hat{k}(\hat{1} + \hat{q}\hat{k})^{-1} \tag{18.21}$$

$$= \hat{k} - \hat{k}(\hat{1} + \hat{q}\hat{k})^{-1}\hat{q}\hat{k} = \hat{k} - \hat{k}\hat{d}\hat{k}. \tag{18.22}$$

The covariance thus becomes

$$\text{Cov}[y(\boldsymbol{x})y(\boldsymbol{x}')] = g(\boldsymbol{x}, \boldsymbol{x}') \tag{18.23}$$

$$= k(\boldsymbol{x}, \boldsymbol{x}') - \sum_{nm} k(\boldsymbol{x}, \boldsymbol{x}_n)(\boldsymbol{C}^{-1})_{nm} k(\boldsymbol{x}_m, \boldsymbol{x}'). \tag{18.24}$$

Chapter 19

Hyperparameters

The prior distribution for the linear basis function models included a so-called hyperparameter, λ, which was considered constant in the probabilistic analysis. The Gaussian process also involves a number of hyperparameters. In the case of the Gaussian kernel, the width of the kernel ℓ and the prefactor k_0 are hyperparameters. If the noise on the data σ is unknown, this can also be regarded as a hyperparameter. In the following we shall refer to the hyperparameters collectively using the symbol $\boldsymbol{\theta}$.

There are at least three different ways to approach hyperparameters. The first one is, if we have some independent prior expectation to the parameters. The parameter ℓ determines the typical length scale of the functions we use, and we might have prior knowledge to estimate a reasonable value. If we are for example interested in the forces between atoms, we expect that they will vary on the length scale of Ångstrøms, and not on the scale of, say, femtometers or kilometers.

The second way to determine hyperparameters is through cross-validation, which was briefly discussed in Chapter 14. In that chapter, we tried different values for the parameter λ and determined the best value through minimization of the error $\int (f(x) - y(x, \boldsymbol{w}_0))^2 \, dx$. This we could of course only do because we knew the function $f(x)$ already. More generally one would construct the model based on a subset of data points (the training set) and then calculate the error on a validation set. This procedure is repeated for different values of the hyperparameters, and the ones that minimizes the error

And finally we have the true Bayesian way! In that case one should in principle assume a prior probability distribution for the hyperparameter $\mathcal{P}(\boldsymbol{\theta})$. Given the data $\boldsymbol{t}$, the posterior distribution for the hyperparameter can then be obtained from Bayes' theorem by marginalization over the model parameters

$$\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta}) = \int \mathcal{P}(\boldsymbol{t}, \boldsymbol{y}|\boldsymbol{\theta}) \, d\boldsymbol{y} \, \mathcal{P}(\boldsymbol{\theta}) \tag{19.1}$$

$$= \int \mathcal{P}(\boldsymbol{t}|\boldsymbol{y})\mathcal{P}(\boldsymbol{y}|\boldsymbol{\theta}) \, d\boldsymbol{y} \, \mathcal{P}(\boldsymbol{\theta}), \tag{19.2}$$

where $\mathcal{P}(\boldsymbol{y}|\boldsymbol{\theta})$ is the prior distribution for the prediction of the data points.

It is seldom possible to work with the full posterior distribution for the hyperparameters, and we shall instead resort to just maximizing the likelihood $\mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta})$. For a Gaussian process, the prior is a Gaussian distribution with the variance given by the

kernel. This means that the prior prediction for the data points are

$$\mathcal{P}(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p, \boldsymbol{K}), \tag{19.3}$$

where $y_p(\boldsymbol{x})$ is the assumed prior average function, and $\boldsymbol{y}_p = (y_p(\boldsymbol{x}_1), y_p(\boldsymbol{x}_2), \ldots, y_p(\boldsymbol{x}_N))$. Furthermore, we assume (as always) that the data points are noisy with variance σ^2:

$$\mathcal{P}(\boldsymbol{t}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \sigma^2 \boldsymbol{I}_N). \tag{19.4}$$

This results in a likelihood (see Appendix 21.1.2 for the details)

$$\mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = \int \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \sigma^2 \boldsymbol{I}_N)\mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p, \boldsymbol{K})\, d\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}), \tag{19.5}$$

where we recover the matrix from before $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$.
The determination of the hyperparameters then consists of maximizing the log-likelihood, which is given by

$$\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = \log \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}) \tag{19.6}$$

$$= -\frac{1}{2}\log\left(\det(\boldsymbol{C})\right) - \frac{1}{2}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{y}_p) - \frac{N}{2}\log(2\pi) \tag{19.7}$$

The maximization can either be done directly with this expression, or by using the gradients, which are given by

$$\frac{\partial}{\partial \theta_i}\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \theta_i}\right) + \frac{1}{2}(\boldsymbol{t}-\boldsymbol{y}_p)^T \boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \theta_i}\boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{y}_p). \tag{19.8}$$

The calculation of the determinant in Eq. 19.7 might be numerically difficult, and it might be better to find the eigenvalues $\alpha_1, \alpha_2, \ldots, \alpha_N$ of $\boldsymbol{C}$, and then use $\log(\det(\boldsymbol{C})) = \sum_{i=1}^{N}\log(\alpha_i)$. This formula is shown by diagonalizing $\boldsymbol{C}$:

$$\boldsymbol{D} = diag(\alpha_1, \alpha_2, \ldots, \alpha_N) = \boldsymbol{U}^{-1}\boldsymbol{C}\boldsymbol{U} \tag{19.9}$$

and then use

$$\det(\boldsymbol{C}) = \det(\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1}) = \det(\boldsymbol{U})\det(\boldsymbol{D})\det(\boldsymbol{U})^{-1} \tag{19.10}$$

$$= \det(\boldsymbol{D}) = \prod_i \alpha_i. \tag{19.11}$$

A simple and interesting case is the determination of a prefactor of $\boldsymbol{C}$, which we therefore write as $\boldsymbol{C} = k_0\boldsymbol{C}_0$. This could for example be the case with no noise ($\sigma = 0$) and with the Gaussian kernel with the prefactor k_0: $k(x, x') = k_0 \exp(-(x-x')^2/2/\ell^2)$. Alternatively, we can write $\boldsymbol{C} = \boldsymbol{K} + \sigma^2\boldsymbol{I}_N = k_0\boldsymbol{K}_0 + \sigma^2\boldsymbol{I}_N = k_0(\boldsymbol{K}_0 + \frac{\sigma^2}{k_0}\boldsymbol{I}_N)$, and consider $\frac{\sigma^2}{k_0}$ as fixed.
We then get

$$\log \mathcal{P}(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{N}{2}\log(k_0) - \frac{1}{2k_0}(\boldsymbol{t}-\boldsymbol{y}_p)^T\boldsymbol{C}_0^{-1}(\boldsymbol{t}-\boldsymbol{y}_p) + \text{const.} \tag{19.12}$$

Calculating the derivative and setting it to zero gives

$$k_0 = \frac{1}{N}(\boldsymbol{t}-\boldsymbol{y}_p)^T\boldsymbol{C}_0^{-1}(\boldsymbol{t}-\boldsymbol{y}_p). \tag{19.13}$$

The prior average function may be a constant or have the form $y_p(\boldsymbol{x}) = \tilde{y}(\boldsymbol{x}) + y_p^0$ with an additive constant y_p^0. This constant can also be treated as a hyperparameter, which can be optimized. The log-likelihood becomes

$$\log \mathcal{P}(\boldsymbol{t}|y_p^0) = -\frac{1}{2}(\boldsymbol{t} - \tilde{\boldsymbol{y}} - y_p^0\boldsymbol{e})^T \boldsymbol{C}^{-1}(\boldsymbol{t} - \tilde{\boldsymbol{y}} - y_p^0\boldsymbol{e}) + \text{constant}, \tag{19.14}$$

where $\tilde{\boldsymbol{y}} = (\tilde{y}(\boldsymbol{x}_1), \tilde{y}(\boldsymbol{x}_2), \ldots, \tilde{y}(\boldsymbol{x}_N))$ and $\boldsymbol{e} = (1, 1, \ldots, 1)$. Setting the derivative to zero we find

$$y_p^0 = \left(\sum_{nm} (\boldsymbol{C}^{-1})_{nm}(t_m - \tilde{y}(x_m))\right) / \left(\sum_{nm} (\boldsymbol{C}^{-1})_{nm}\right). \tag{19.15}$$

Exercise: Bayesian determination of hyperparameters

Take another look at the sine fitting problem with the Gaussian kernel. Plot the log-likelihood Eq. 19.7 as a function of ℓ and k_0. (You can for example make a contour plot). What are the optimal values?

Chapter 20

Example from materials science: Perovskites

.
We end this note with a realistic research project using machine learning. The aim is to use a Gaussian process to predict the heat of formation of perovskites. The exercise is based on the work in Refs. [Cas+12a] and [Cas+12b].

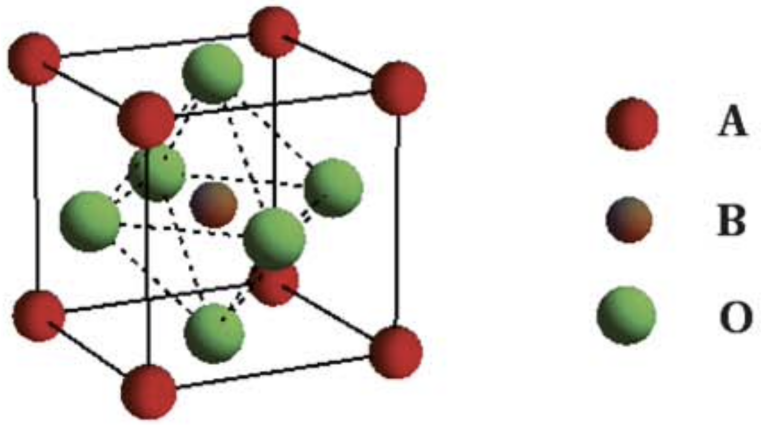

Figure 20.1: The crystal structure of a cubic perovskite

Perovskites is a large group of materials with the composition ABX_3, where A and B are cations (positive valences) and X is an anion. In the cubic phase, which is the only one we shall look at here, the crystal structure looks as shown on Figure 20.1. As A and B we consider 52 different metallic elements, and X_3 are different combinations of O, N, S, and F, for example $X_3 = O_2N$. The metallic elements are shown in the periodic table Figure 20.2.
The aim of the work in Refs. [Cas+12a] and [Cas+12b] was to identify new materials, which can absorb light and be used in solar cells and water splitting devices. The primary properties of importance of these materials are their stabilities and their band gaps. The stability is important because unstable materials quickly degrade, and the band gap should be in the range $1\,\mathrm{eV}$ to $3\,\mathrm{eV}$ so that visible light can be absorbed. Here we shall concentrate on the stability through determination of the heat of formation, E,

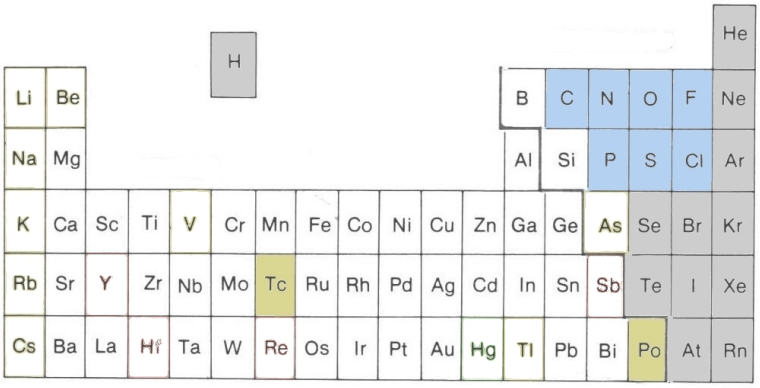

Figure 20.2: The 52 metallic elements used in this study are in white and to the left of the zig-zag line.

relative to competing sub-materials. E is in other words the energy of the perovskite minus the energy of the most stable combination of other materials made from the same atoms. For example, $CaTiO_3$ could in principle disintegrate into CaO and TiO_2, but it does not do that because the energy of $CaTiO_3$ is lower corresponding to a negative E. We are therefore interested in identifying perovskites with negative heats of formation. Because of the limited accuracy of the DFT calculations, we shall make the window a little wider, so we are interested in materials with $E < 0.2\,\mathrm{eV}$.

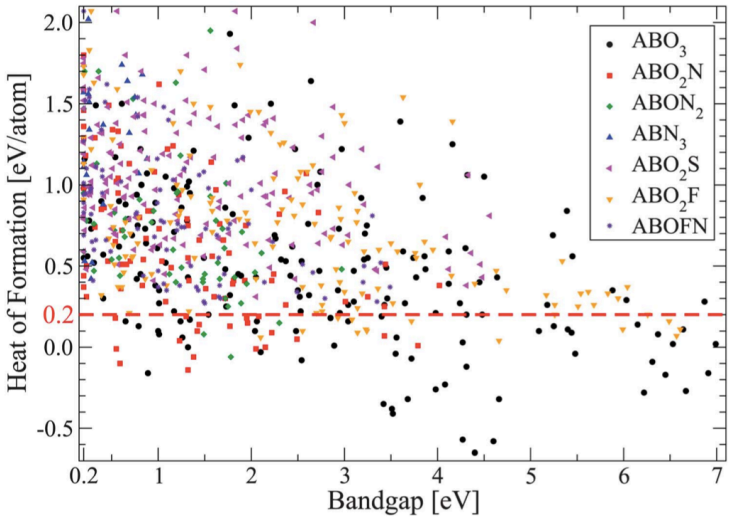

Figure 20.3: The calculated band gaps and heat of formation for $\sim$ 20000 perovskites. The interesting materials are those with $E < 0.2$ and the band gap in the range $1\,\mathrm{eV}$ to $3\,\mathrm{eV}$

Figure 20.3 shows the calculated band gaps and heats of formation for $\sim$ 20000 perovskites. The calculations have been performed with density functional theory (DFT), which is a fairly computer-intensive quantum mechanical approach. We see that only rather few materials (maybe ~ 50) are in the interesting region, and it therefore seems

a waste of computer time to calculate the other more than 19000 compounds. So the question is: *Can we use machine learning, so that we can identify the perovskites with negative heats of formation with fewer DFT calculations?*
We shall use a Gaussian process with a Gaussian kernel to model the heats of formation and we therefore need to represent the materials to the kernel. This is done with so-called feature vectors.

20.1 Feature vectors

The Gaussian kernel has the form

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_0 \exp\left(-|\boldsymbol{x} - \boldsymbol{x}'|^2/2\ell^2\right), \tag{20.1}$$

where $\boldsymbol{x}$ (and $\boldsymbol{x}'$) is a vector which should represent the material. Such a vector is called a *feature vector* or a *descriptor* or sometimes a *fingerprint*.
The purpose of the kernel is to identify materials, which are "close" in the sense that they have similar properties. The feature vector should therefore be constructed so that $|\boldsymbol{x} - \boldsymbol{x}'|$ is small if the materials are similar.
Our materials are identified by the atoms A, B and up to two different elements in X_3, and this information should somehow be turned into a vector.
One way of doing this is to use so-called *one-hot encoding*. (We shall in fact not use this, but it is worth knowing!) We shall illustrate this with encoding of the atom A. In our study A can take on 52 different values corresponding to the metallic elements, in alphabetical order: Ag, Al, As and so on. The one-hot encoding consists in making a 52 dimensional vector with all components equal to zero except for one. The encoding of Ag is $(1, 0, 0, 0, \ldots, 0)$; the encoding of Al is $(0, 1, 0, 0, \ldots, 0)$; As becomes $(0, 0, 1, 0, \ldots, 0)$ and so on. The advantage of this is that it is simple; the disadvantage is that the machine do not learn much across elements. Two materials with the same A atom will have distance 0 (for the A-part, we still need to encode B and X_3), while two materials with different A element will have distance 2. But we know that some atoms are more similar than others! Pd and Pt are for example chemically rather similar because the belong to the same row in the periodic table. An alternative encoding, which is the one that we shall use, is therefore to represent atom A with its coordinates in the periodic table. So now Ag becomes $(5, 11)$; Al becomes $(3, 13)$; As becomes $(4, 15)$ etc. With this encoding the distance between to feature vectors is now the distance in the periodic table!
We shall encode B in a similar way and just put the two (two-dimensional) vectors for A and B after each other so we get a vector with 4 components.
What about X_3? X can take on four values O, N, S, and F and these we shall *count-encode*. We make a vector with four components, the first one being the number of O atoms, the second component being the number of N atoms etc. ON_2 is in this way encoded as $(1, 2, 0, 0)$. This vector is now put at the end of the A+B vector and we end up with an 8-dimensional feature vector, which we can use in the Gaussian kernel.

Exercise: Heat of formation of perovskites

At DTU Learn you can find an ASE database, which contains all the data and a short Jupyter notebook to read in the data. As you may note the heats of

formation have only two decimal places, so an estimate of the noise could be $\sigma = 0.01/(2\sqrt{3}) \sim 0.003$ eV. This approximates the round-off distribution with a Gaussian with the same variance.
So now you are ready to play with the Gaussian process.
Some questions:

- How well can you predict the DFT energies?
- How few DFT calculations can you get away with if you want to know E within $0.5\,\mathrm{eV}$?
- Does the predicted uncertainty agree with the real error on the average?
- What are reasonable hyperparameters k_0 and ℓ? Can you determine them with cross validation? With the Bayesian approach.
- Can you find a strategy for which DFT calculations to perform if we want to identify the ones with low heats of formation?
- Find more relevant questions and try to answer them!

Part IV

Appendices

Chapter 21

Useful formulas for probability distributions

In general you can find a very systematic list of probability distributions and their properties at Wikipedia.

21.1 Gaussian distribution

The probability density function is

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(x-\mu)^2/2\sigma^2}, \tag{21.1}$$

and we have

$$\langle x \rangle = \mu \text{ and } \text{Var}[x] = \sigma^2 \tag{21.2}$$

For a set of data $x_1, x_2, \ldots x_N$

$$\prod_i \mathcal{N}(x_i|\mu,\sigma^2) \propto \mathcal{N}(\bar{x}|\mu,\sigma^2/N) \cdot \mathcal{N}(\overline{\Delta x^2}|0,\sigma^2/N), \tag{21.3}$$

with $\bar{x} = \sum_i x_i/N$ and $\overline{\Delta x^2} = \sum_i (x_i - \bar{x})^2/N$.

21.1.1 Multi-dimensional Gaussian distribution

If the variable is a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$ we can define a multi-dimensional Gaussian distribution as

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{21.4}$$

where Σ is a symmetric and positive definite $D \times D$ matrix and $\mu = (\mu_1, \mu_2, \ldots, \mu_D)$ is a D-dimensional vector.
The average and the covariance matrix is given by

$$\langle \boldsymbol{x} \rangle = \boldsymbol{\mu} \text{ and } \text{Cov}[\boldsymbol{x}, \boldsymbol{x}^T] = \boldsymbol{\Sigma} \tag{21.5}$$

Let us take a closer look at how this comes about, including why the determinant appears in the prefactor. Let us start by diagonalizing the matrix $\boldsymbol{\Sigma}$. Since all the eigenvalues are positive, we can write them as $(\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2)$. If $\boldsymbol{U}$ is the unitary matrix diagonalizing $\boldsymbol{\Sigma}$, we have

$$(\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2) = \boldsymbol{U}^T \boldsymbol{\Sigma} \boldsymbol{U}. \tag{21.6}$$

and therefore

$$\boldsymbol{D} \equiv \left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \ldots, \frac{1}{\sigma_D^2} \right) = \boldsymbol{U}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{U} \tag{21.7}$$

If we introduce new variables by $\boldsymbol{z} = \boldsymbol{U}^T(\boldsymbol{x} - \boldsymbol{\mu})$ we can write the distribution

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \exp\left(-\frac{1}{2} \boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z} \right) = \prod_i \exp(-z_i^2/2\sigma_i^2). \tag{21.8}$$

We see that using the variables z_i, the probability distribution is a product of *independent* distributions in $z_1, z_2, \ldots, z_D$. Knowing the prefactor for the one-dimensional Gaussian distribution (see Eq. 21.1), the prefactor for the multi-dimensional Gaussian therefore becomes

$$\prod_{i=1}^{D} \frac{1}{(2\pi\sigma_i^2)^{1/2}} = \frac{1}{(2\pi)^{D/2}} \det(\boldsymbol{D})^{1/2} = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det(\boldsymbol{\Sigma})^{1/2}}, \tag{21.9}$$

where the last equality sign comes about because the determinant of a product of matrices is the product of determinants, and $\det \boldsymbol{U} = 1$.
The coordinate transformation also makes it straightforward to calculate the average

$$\langle \boldsymbol{x} \rangle = \langle \boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{z} \rangle = \mu + \boldsymbol{U} \langle \boldsymbol{z} \rangle = \boldsymbol{\mu}, \tag{21.10}$$

and the covariance matrix

$$\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{x}^T] = \langle (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T \rangle = \langle \boldsymbol{U}\boldsymbol{z}\boldsymbol{z}^T\boldsymbol{U}^T \rangle = \boldsymbol{U} \langle \boldsymbol{z}\boldsymbol{z}^T \rangle \boldsymbol{U}^T \tag{21.11}$$

$$= \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T = \boldsymbol{\Sigma}. \tag{21.12}$$

21.1.2 Convolution of multi-dimensional Gaussian distributions

In some situations several multi-dimensional normal distribution have to be combined. Imagine, that the variable $\boldsymbol{y}$ is Gaussianly distribution $\mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p, \boldsymbol{K})$ and then another variable $\boldsymbol{t}$ is distributed around $\boldsymbol{y}$ as $\mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \boldsymbol{\Sigma})$. The marginal distribution is then

$$\mathcal{P}(\boldsymbol{t}) = \int \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \boldsymbol{\Sigma}) \mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_p, \boldsymbol{K})\, d\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}_p, \boldsymbol{C}), \tag{21.13}$$

where $\boldsymbol{C} = \boldsymbol{K} + \boldsymbol{\Sigma}$.

21.1.3 Product of multi-dimensional Gaussian distributions

In Chapter 9 we discussed that the product of two one-dimensional Gaussian distributions is (proportional to) a new Gaussian distribution. This also holds in higher dimensions. The general formula is

$$\mathcal{N}(\boldsymbol{y}|\tilde{\boldsymbol{y}}, \boldsymbol{K}) \propto \mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_1, \boldsymbol{K}_1) \mathcal{N}(\boldsymbol{y}|\boldsymbol{y}_2, \boldsymbol{K}_2), \tag{21.14}$$

where

$$\boldsymbol{K}^{-1} = \boldsymbol{K}_1^{-1} + \boldsymbol{K}_2^{-1} \text{ and } \tilde{\boldsymbol{y}} = \boldsymbol{K} \left(\boldsymbol{K}_1^{-1} \boldsymbol{y}_1 + \boldsymbol{K}_2^{-1} \boldsymbol{y}_2. \right) \tag{21.15}$$

21.2 Binomial distribution

The binomial distribution is a function of an integer $n = 0, 1, 2 \ldots N$ with a probability parameter p:

$$\mathcal{P}(n|N,p) = \binom{N}{n} p^n (1-p)^{(N-n)}, \tag{21.16}$$

where the prefactor is the binomial coefficient

$$\binom{N}{n} = \frac{N!}{n!(N-n)!} = \frac{\Gamma(N+1)}{\Gamma(n+1)\Gamma(N-n+1)}, \tag{21.17}$$

where $\Gamma(z)$ denotes the gamma function (see more below in the section about the beta distribution). The binomial distribution obeys

$$\langle n \rangle = pN \text{ and } \text{Var}[n] = Np(1-p). \tag{21.18}$$

21.3 Beta distribution

The probability density function is

$$\mathcal{B}(x|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{21.19}$$

where

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}. \tag{21.20}$$

Γ is the gamma function $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx$. (If you do not already know the gamma function, it is like a continuous version of the factorial function. For integers, n, we have $\Gamma(n) = (n-1)!$.)
The beta distribution obeys

$$\langle x \rangle = \frac{\alpha}{\alpha+\beta} \text{ and } \text{Var}[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{21.21}$$

21.4 Dirichlet distribution

The probability density function is

$$\mathcal{D}(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \left(\prod_n x_n^{\alpha_n - 1} \right) \delta(\sum_n x_n - 1), \tag{21.22}$$

with

$$B(\boldsymbol{\alpha}) = \frac{\prod_n \Gamma(\alpha_n)}{\Gamma(\alpha)}, \text{where } \alpha = \sum_n \alpha_n \tag{21.23}$$

The averages and covariances are

$$\langle x_n \rangle = \frac{\alpha_n}{\alpha} \text{ and } \text{Cov}[x_n, x_m] = \frac{1}{(\alpha+1)} \left(\delta_{nm} \frac{\alpha_n}{\alpha} - \frac{\alpha_n \alpha_m}{\alpha^2} \right) \tag{21.24}$$

Part V

Solutions to some exercises

The extended sum rule

Use that $X \vee Y = \overline{\overline{X} \wedge \overline{Y}}$ and apply both the sum rule and the product rule.

Extended, extended sum rule

Use the extended sum rule with $X = Y_1$ and $Y = Y_2 \wedge Y_3 \ldots Y_n$. Continue with $X = Y_2$ and $Y = Y_3 \wedge Y_4 \ldots Y_n$ and so on.

Independence of propositions

Use the extended sum rule.

Girls or boys

1. They have at least one girl ("$\geq$1g")

$$\mathcal{P}(\text{2g}|\geq \text{1g}) = \frac{\mathcal{P}(\geq \text{1g}|\text{2g})\mathcal{P}(\text{2g})}{\mathcal{P}(\geq \text{1g})} = \frac{\mathcal{P}(\text{2g})}{\mathcal{P}(\geq \text{1g})} = \frac{1/4}{3/4} = \frac{1}{3} \tag{21.25}$$

2. The oldest child is a girl ("old")

$$\mathcal{P}(\text{2g}|\text{old}) = \frac{\mathcal{P}(\text{old}|\text{2g})\mathcal{P}(\text{2g})}{\mathcal{P}(\text{old})} = \frac{\mathcal{P}(\text{2g})}{\mathcal{P}(\text{old})} = \frac{1/4}{1/2} = \frac{1}{2} \tag{21.26}$$

3. They have a girl with blue eyes ("blue") Denote the probability of blue eyes with p.

$$\mathcal{P}(\text{2g}|\text{blue}) = \frac{\mathcal{P}(\text{blue}|\text{2g})\mathcal{P}(\text{2g})}{\mathcal{P}(\text{blue})} = \frac{\mathcal{P}(\text{blue}|\text{2g})\mathcal{P}(\text{2g})}{\mathcal{P}(\text{blue}|\text{2g})\mathcal{P}(\text{2g}) + \mathcal{P}(\text{blue}|\text{1g})\mathcal{P}(\text{1g})} \tag{21.27}$$

$$= \frac{\left(p^2 + 2p(1-p)\right) \cdot 1/4}{(p^2 + 2p(1-p)) \cdot 1/4 + p \cdot 1/2} = \frac{2-p}{4-p} \tag{21.28}$$

   So if $p \sim 1/2$, $\mathcal{P}(\text{2g}|\text{blue}) \sim 3/7$.

4. They have a girl with the name "Mushroom". Same analysis as for blue eyes, but now $p \ll 1$, so $\mathcal{P}(\text{2g}|\text{Mushroom}) \sim \frac{1}{2}$.

Medical screening

$$\begin{aligned} \mathcal{P}(ill|pos) &= \frac{\mathcal{P}(pos|ill)\mathcal{P}(ill)}{\mathcal{P}(pos)} = \frac{\mathcal{P}(pos|ill)\mathcal{P}(ill)}{\mathcal{P}(pos|ill)\mathcal{P}(ill) + \mathcal{P}(pos|h)\mathcal{P}(h)} \\ &= \frac{0.99 \cdot 0.001}{0.99 \cdot 0.001 + 0.01 \cdot 0.999} \sim 0.09 \end{aligned} \tag{21.29}$$

The result depends significantly on the prior probability (also sometimes called the base probability). If the frequency of the disease in the population is 1%, the probability of being ill given a positive test becomes 0.5.

Newton's second law

At the outset the prior probability $\frac{\mathcal{P}(\mathcal{M})}{\mathcal{P}(\tilde{\mathcal{M}})}$ must be considered quite small. If somebody just postulated Newton's second law it wouldn't be believable. However, the likelihood $\frac{\mathcal{P}(\mathcal{D}|\mathcal{M})}{\mathcal{P}(\mathcal{D}|\tilde{\mathcal{M}})}$ is large since there are so many examples of experiments, which can be explained by Newton's second law. If the product of the two ends up being large, we get a final probability close to one.

From maximum entropy to the Gaussian distribution

This is an optimization problem with constraints. We use the Lagrange multiplier method, and shall therefore maximize

$$\tilde{S} = S - \lambda_\mu \langle x \rangle - \lambda_\sigma \text{Var}[x] = -\int \mathcal{P}(x) \left(\log(\mathcal{P}(x)) + \lambda_\mu x + \lambda_\sigma x^2 \right) \, dx, \quad (21.30)$$

with respect to the probability distribution $\mathcal{P}(x)$, and use the constraints. Varying the probability distribution by $\delta\mathcal{P}$ we get

$$\delta\tilde{S} = -\int \delta\mathcal{P}(x)(\log(\mathcal{P}(x)) + \lambda_\mu x + \lambda_\sigma x^2 - 1) \, dx = 0. \quad (21.31)$$

(The "1" comes from the derivative of the logarithm). Solving the equation gives

$$\mathcal{P}(x) = Ae^{1-\lambda_\mu x - \lambda x^2}, \quad (21.32)$$

and using the constraints, the λ's are determined so we get

$$\mathcal{P}(x) = \frac{1}{(2\pi\sigma^2)^{(1/2)}} e^{-(x-\mu)^2/2\sigma^2}. \quad (21.33)$$

Gaussianly distributed data

$$\mathcal{P}(\mu|x_1, x_2, \ldots x_N, \sigma^2) \propto \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-((\mu - \bar{x})^2 + \overline{\Delta x^2})/2\sigma_N^2 \right), \quad (21.34)$$

with $\sigma_N^2 = \sigma^2/N$.
Optimal value is when $\mu = \bar{x}$.
The posterior distribution is seen to be a normal distribution $\mathcal{N}(\mu|\bar{x}, \sigma_N)$. The variance of the average value is thus $\sigma_N^2 = \sigma^2/N$, so the uncertainty on the average value decays as $1/\sqrt{N}$.

Tversky and Kahneman

The smaller hospital. We saw in the exercise above that the uncertainty on the average decays as $1/\sqrt{N}$. Small samples tend to have large fluctuations. If a very tiny hospital had only one delivery per day, it would be over 60% boys half of the days!

Playing with priors – the average height of Norwegians

According to Bayes' theorem and the product rule for Gaussians we have

$$\mathcal{P}(h_{\text{Nor}}|h_1, h_2 \dots h_N) \propto \left(\prod_n \mathcal{N}(h_n|h_{\text{Nor}}, \sigma^2_{\text{Nor}}) \right) \mathcal{N}(h_{\text{Nor}}|h_{\text{DK}}, \sigma^2_d) \tag{21.35}$$

$$\propto \mathcal{N}(h_{\text{Nor}}|\bar{h}, \sigma^2_{\text{Nor}}/N)\, \mathcal{N}(h_{\text{Nor}}|h_{\text{DK}}, \sigma^2_d) \tag{21.36}$$

$$\propto \mathcal{N}(h_{\text{Nor}}|h_0, \sigma^2_0), \tag{21.37}$$

with

$$h_0 = \frac{\frac{\bar{h}}{\sigma^2_{\text{Nor}}/N} + \frac{h_{\text{DK}}}{\sigma^2_d}}{N/\sigma^2_{\text{Nor}} + 1/\sigma^2_d} \tag{21.38}$$

$$= \frac{\sum_i h_i + (\frac{\sigma_{\text{Nor}}}{\sigma_d})^2 h_{\text{DK}}}{N + (\frac{\sigma_{\text{Nor}}}{\sigma_d})^2}. \tag{21.39}$$

We see that the result corresponds to adding $\sigma^2_{\text{Nor}}/\sigma^2_d = 4$ Danish average heights when calculating the mean value.
The width of the posterior distribution is given by

$$\sigma^2_0 = \frac{\sigma^2_{\text{Nor}}}{N + (\frac{\sigma_{\text{Nor}}}{\sigma_d})^2}. \tag{21.40}$$

As with the Gaussian data example above, we can see that the width is decaying as $1/\sqrt{N}$, where the Danes enter with a "count" of $(\frac{\sigma_{\text{Nor}}}{\sigma_d})^2$. If N is large, the prior does not matter for the estimates.

Is the coin fair?

The posterior becomes

$$\mathcal{P}(p) = \mathcal{B}(p|\alpha = N_h + 1, \beta = N_t + 1), \tag{21.41}$$

with the average (see appendix about the beta distribution)

$$\langle p \rangle = \frac{\alpha}{\alpha + \beta} = \frac{N_h + 1}{N_h + N_t + 2} \tag{21.42}$$

The result corresponds to calculating the "usual" average but with one extra pseudo count for heads and one for tails.
The uncertainty is the square root of the variance

$$\text{Var}[p] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} = \frac{(N_h + 1)(N_t + 1)}{(N + 2)^2(N + 3)} \tag{21.43}$$

If $N_h \sim N_t \sim N/2 \gg 1$ we get $\text{Var}[p] \sim (N/2)^{-1}$ as might be expected.
If we get 6 heads out of 10, the average is $\langle p \rangle = 7/12 = 0.58$ and the width is $\sqrt{\text{Var}[p]} = 0.14$, so 0.58 ± 0.14 means that the coin could be fair. With 600 heads out

of 1000, we get $\langle p \rangle = 0.60$ and the width $\sqrt{\text{Var}[p]} = 0.02$, so with 0.60 ± 0.02 your friend is most likely cheating.
If we use a prior $\mathcal{P}_0(p) = \mathcal{B}(p|\alpha_0, \alpha_0)$, we get for the posterior $\mathcal{B}(N_h + \alpha_0, N_t + \alpha_0)$, so the average becomes

$$\langle p \rangle = \frac{N_h + \alpha_0}{N_h + N_t + 2\alpha_0}, \tag{21.44}$$

corresponding to a pseudo count of α_0 for both heads and tails. The constant prior ($\mathcal{P}_0(p) = 1$) corresponds to $\alpha_0 = 1$. One can see that larger values of α_0 drives the average result towards 1/2, expressing your prior stronger trust in your friend.

But *is* the coin really fair?

We need to calculate the likelihoods for the two models. Let us start with model $\mathcal{M}_1$. According to Bayes' theorem we need to evaluate

$$\mathcal{P}(\mathcal{D}|\mathcal{M}_1) = \int_0^1 \mathcal{P}(\mathcal{D}|\mathcal{M}_1, p)\mathcal{P}(p|\mathcal{M}_1)\, dp = \int_0^1 \mathcal{P}(\mathcal{D}|\mathcal{M}_1, p)\, dp \tag{21.45}$$

$\mathcal{P}(p|\mathcal{M}_1)$ is the prior for p given the model, and here we have just taken that to be unity. The likelihood is a binomial distribution, and we can see that we have to integrate that over p. There are several ways to do that. Here we use that we can relate the binomial distribution to the beta distribution (see the appendix):

$$\mathcal{P}_{bin}(N_h|N, p) = \binom{N}{N_h} p^{N_h}(1-p)^{N_t} \tag{21.46}$$

$$= \binom{N}{N_h} B(N_h + 1, N_t + 1)\mathcal{B}(p|N_t + 1, N_h + 1), \tag{21.47}$$

where $B(N_h + 1, N_t + 1)$ is the beta function. The trick here is, that the beta distribution is normalized, so the integral over p of that term is 1. Writing out the binomial coefficient and the beta function in terms of gamma functions (or factorials, if you like) gives in the end

$$\mathcal{P}(\mathcal{D}|\mathcal{M}_1) = \frac{1}{N+1}. \tag{21.48}$$

At least the end result of this part of the calculation was simple!
And now for model $\mathcal{M}_2$. Since there is no integration over a parameter here, we just get

$$\mathcal{P}(\mathcal{D}|\mathcal{M}_2) = \binom{N}{N_h} (\frac{1}{2})^{N_h} (\frac{1}{2})^{N_t}. \tag{21.49}$$

This is a little bit ugly because of the factorials in the binomial coefficient, but for reasonably large values of N, these can be approximated using Stirling's formula: $\log(N!) \sim N\log(N) - N$. An equivalent and more direct way is to use the Gaussian approximation for the binomial distribution, and this gives

$$\mathcal{P}(\mathcal{D}|\mathcal{M}_2) \sim \frac{1}{N}\mathcal{N}(x = N_h/N|p, \sigma^2) = \frac{1}{N}\mathcal{N}(x = N_h/N|1/2, \sigma^2), \tag{21.50}$$

with $\sigma^2 = p(1-p)/N = 1/4N$.

So now we are ready to compare the two models. Let us set the prior for the models to one:

$$\frac{\mathcal{P}(\mathcal{M}_2)}{\mathcal{P}(\mathcal{M}_1)} = 1. \tag{21.51}$$

(You can put in a different number, as a prior evaluation of your friend's trustworthiness, for example from previous experience.)
Model $\mathcal{M}_2$ is preferred (*i.e.* we can consider the coin as fair) if

$$\frac{\mathcal{P}(\mathcal{D}|\mathcal{M}_2)}{\mathcal{P}(\mathcal{D}|\mathcal{M}_1)} > 1. \tag{21.52}$$

With a bit of rewriting (and for example setting $(N+1)/N \sim 1$) this amounts to

$$|x - \frac{1}{2}| < \sigma\sqrt{|\log(2\pi\sigma^2)|}, \tag{21.53}$$

with $x = N_h/N$ and where as before, we have $\sigma^2 = 1/4N$. (We use $\log(x)$ to describe the natural logarithm with base e.)
We see that except for the logarithmic term we should compare the deviation of N_t/N from $1/2$ to the width σ to determine if the coin is fair. This is pretty similar to the conclusion above, where we only looked at model $\mathcal{M}_1$. We also see that as N increases, N_t/N has to be closer and closer to $1/2$ to conclude that the coin is fair.
What about the logarithmic term? In fact it doesn't matter much. If we write the number of experiments as $N = 10^\gamma$, the logarithmic term becomes $\sqrt{2.30\gamma - 0.45}$, so if we have 10 experiments the square-root-log-factor is 1.4, while with 10^5 experiments it becomes 3.3. Not a big change considering that σ at the same time varies by a factor of 100. For the example with 6 heads out of 10 experiments, we get $|x - 1/2| = 0.1 < 0.21$, so we may regard the coin as fair. For 600 out of 1000, we get $|x - 1/2| = 0.1 > 0.04$, so the coin is hardly fair.

Course evaluations at DTU

The likelihood is a multinomial distribution

$$\mathcal{P}(\boldsymbol{N}|\boldsymbol{p}) = \frac{N!}{N_1!\,N_2!\,N_3!\,N_4!\,N_5!} p_1^{N_1} p_2^{N_2} p_3^{N_3} p_4^{N_4} p_5^{N_5} \tag{21.54}$$

and the posterior distribution therefore becomes a Dirichlet distribution (see appendix)

$$\mathcal{P}(p_1, \ldots p_5)\mathcal{D}(\boldsymbol{p}|\boldsymbol{\alpha} = \boldsymbol{N} + 1), \tag{21.55}$$

where $\boldsymbol{N} = (N_1, N_2, N_3, N_4, N_5)$ are the number of student "votes". The final grade is $g = \sum_n np_n$ with the average value

$$\langle g \rangle = \sum_n n\langle p_n \rangle = \sum_n n\frac{N_n + 1}{N + 5}. \tag{21.56}$$

We see that the average final grade is obtained by adding a pseudo count of one to each of the five grades. In the case of the course on Quantum Transport Theory with 4 students all voting 5, we will therefore now find an average final grade of $\langle g \rangle =$

$(1\cdot 1+1\cdot 2+1\cdot 3+1\cdot 4+5\cdot 5)/9 = 3.9$. Because of the low number of students, the uncertainty is large. It is given by the variance of the final grade, which can be found from the covariance of the Dirichlet distribution (see appendix):

$$\text{Var}[g] = \frac{1}{(\alpha+1)}\left(\sum_n n^2\frac{\alpha_n}{\alpha} - \left(\sum_n n\frac{\alpha_n}{\alpha}\right)^2\right), \tag{21.57}$$

where $\alpha = \sum_n \alpha_n$.
With $\boldsymbol{\alpha} = \boldsymbol{N} + 1$ we get in the case of the course on Quantum Transport Theory an uncertainty

$$\sigma = \sqrt{\text{Var}[g]} = 0.5 \tag{21.58}$$

So the final estimate for the average grade is 3.9 ± 0.5.

Regularized least squares method

The error versus the value of the regularization parameter $\tilde{\lambda}$ should look like Figure 21.1.

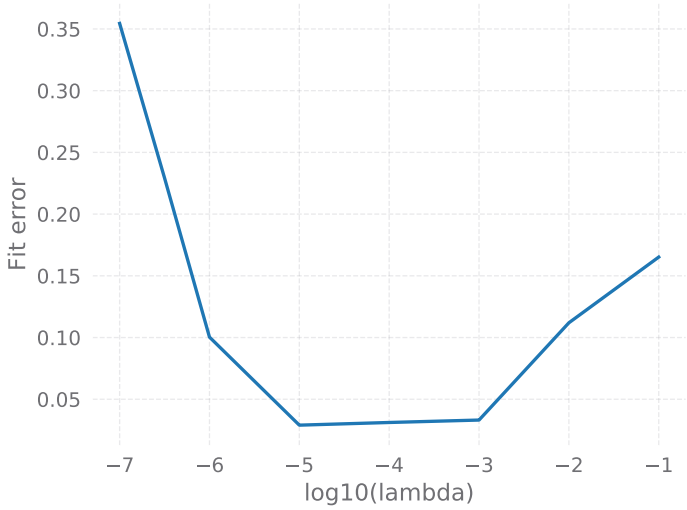

Figure 21.1: The error as a function of $\log_{10}(\tilde{\lambda})$.

It seems that $\tilde{\lambda} \sim 10^{-4}$ is a good choice. This correspond to $\lambda = \tilde{\lambda}/\sigma^2 = 0.01$ or a prior probability distribution of

$$\mathcal{P}(\boldsymbol{w}) \propto e^{-\frac{1}{2}\sum_i |w_i|^2/(10)^2}, \tag{21.59}$$

which seems quite reasonable.

Bayesian analysis of a linear basis function model

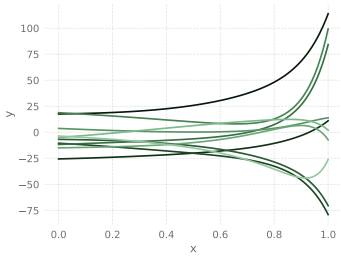

Figure 21.2: An ensemble of functions drawn from the prior distribution Eq. 14.1. We see that the region around $x = 0$ is treated quite differently than the region around $x = 1$. This is due to the polynomial basis functions, where only the zeroth order polynomial is non-vanishing at $x = 0$.

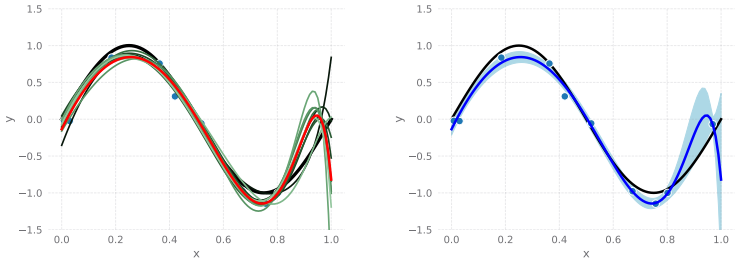

Figure 21.3: Left: An ensemble of functions from the posterior distribution. The thick black line is the sine function. The red line is the optimal fit. Right: The sine function in black and the best fit function in blue. The estimated error bar (plus/minus the square root of the variance) is shown as the shaded area.

Figure 21.2 shows an ensemble of functions drawn from the prior distribution with $\lambda = \tilde{\lambda}/\sigma^2 = 0.01$. Because of the polynomial expansion the functions look quite different close to 1 compared to close to 0, so this distribution may not be a good starting point.

Figure 21.3 (left) shows an ensemble of functions drawn form the posterior distribution. It is clear that the average function is fairly wrong close to 1, but this can also be seen in the ensemble. The ensemble thus gives a "warning", that the fit is not very well determined here.

Figure 21.3 (right) shows the average function together with the uncertainty. Again the error is large close to $x = 1$, but we are warned about this because of the large uncertainty. Also note, that close to the last data point close to $x = 0.97$ the uncertainty is suppressed because of the information from the data point.

Kernel regression

Figure 21.4 (left) illustrates the use of kernel regression in the case, where there is no noise on the data points. Results are shown for $\ell = 0.01, 0.1, 0.5$ and 1.0. For $\ell = 0.01$

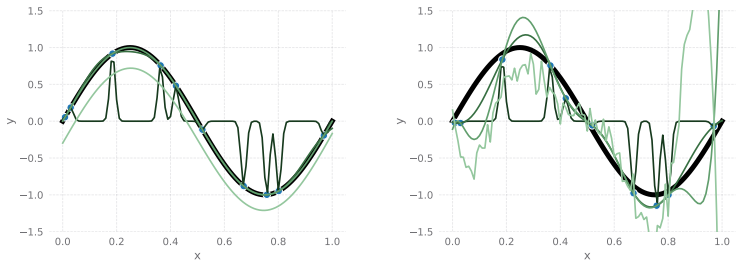

Figure 21.4: Left: Result of kernel regression on data without noise. Right: Result of kernel regression on data with noise.

the Gaussians are localized at each point leading to the spiky curve. For $\ell = 0.1, 0.5$ the fits are great, but for $\ell = 1.0$, the matrix G is close to singular, and the quality of the fit degrades due to numerical problems.
Figure 21.4 (right) shows the results for the same parameters $\ell = 0.01, 0.1, 0.5$ and 1.0, but now with noisy data. The noise clearly makes the fitting much more difficult leading to worse fits and serious numerical problems for large values of ℓ.

Kernel ridge regression

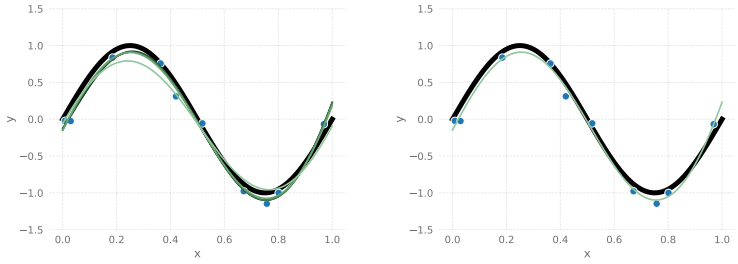

Figure 21.5: Left: Fits produced with kernel ridge regression for $\ell = 0.5$ and λ' in the range $10^{-6} - 10^{-2}$. Only for the highest value $\lambda' = 10^{-2}$ (the brightest curve), the regularization is so strong that the model begins to exhibit high bias. Right: The best fit obtained with kernel ridge regression with the parameters $\ell = 0.5$ and $\lambda' = 10^{-4}$.

Figure 21.6 shows the fit error as a function of the parameter λ' for $\ell = 0.5$. A value of $\lambda' \sim 10^{-4}$ seems appropriate.

Prior distribution with Gaussian kernel function

Figure 21.7 shows ensembles from the prior distribution using a Gaussian kernel function. The length parameter ℓ is set to 0.2 and 0.6 in the two figures, respectively. The parameter is seen to control the typical length scale for the functions in the prior probability distribution.

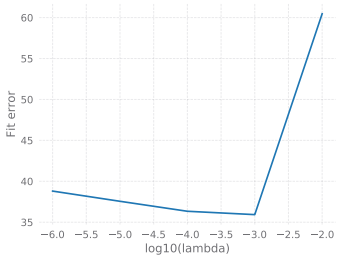

Figure 21.6: Fit error as a function of the parameter λ' for $\ell = 0.5$.

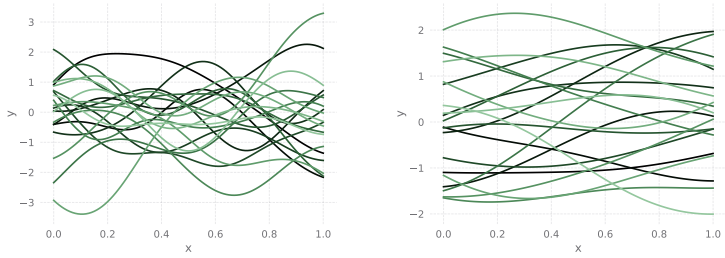

Figure 21.7: Left: An ensemble of functions from the prior distribution using the Gaussian kernel with $\ell = 0.2$. Right: An ensebmle of functions from the prior distribution using the Gaussian kernel with $\ell = 0.6$.

Gaussian process for the sine function

Figure 21.8 shows the results of the Gaussian process for the sine problem. The scale $\ell = 0.05$ is clearly to short leading to overfitting and large variance. With the scale $\ell = 0.5$ significant bias is observed. The uncertainty estimate is reasonable except in the last case, where the error is underestimated.

If the noise parameter is set to zero, the matrix $\boldsymbol{K}$ may become singular depending on the applied length scale. If, for example, $\ell = 0.5$, the eigenvalues of $\boldsymbol{K}$ span over 12 orders of magnitude, and is therefore very difficult to treat numerically. Adding the noise, $\boldsymbol{C} = \boldsymbol{K} + \sigma^2 \boldsymbol{I}_N$, makes $\boldsymbol{C}$ much better behaved since the noise effectively removes all the smallest eigenvalues.

Bayesian determination of hyperparameters

The hyperparameters are determined by maximizing the log-likelihood Eq. 19.7. Figure 21.9 shows the log-likelihood as a function of the two parameters ℓ and k_0 for a fixed value of $\sigma = 0.1$. In our case we know the noise in advance. If that was not the case σ could be considered a hyperparameter as well. The maximum of the log-likelihood is obtained at $\ell = 0.18$ and $k_0 = 0.46$.

Heat of formation of perovskites

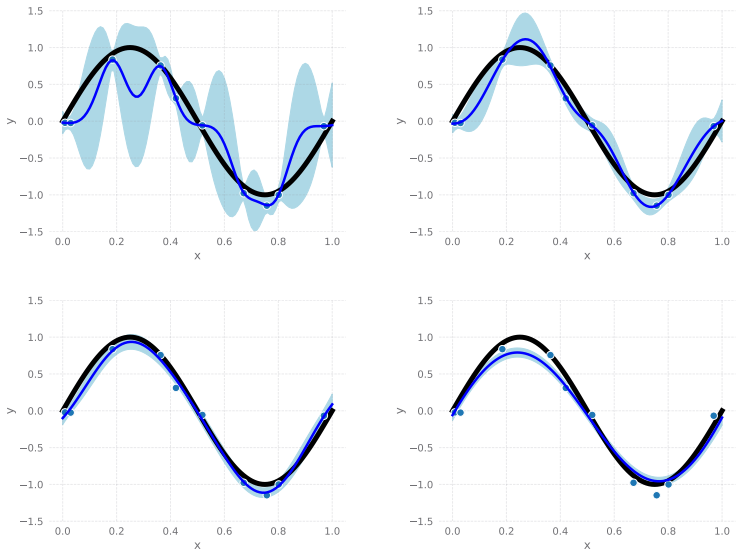

Figure 21.8: Gaussian process regression for the sine function with paramaters $\sigma = 0.1$, $k_0 = 1$. The length parameter has the values $\ell = 0.05$ (upper left), $\ell = 0.1$ (upper right), $\ell = 0.2$ (lower left), and $\ell = 0.5$.

This solution can be studied in more detail in the attached Jupyter notebook.
We use a Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_0 \exp(-(\boldsymbol{x} - \boldsymbol{x}')^2/2\ell^2), \tag{21.60}$$

where $\boldsymbol{x}$ denotes the fingerprint for a material as described above. We write the noise as $k_0\sigma^2$, with $\sigma = 0.003$. In this way k_0 becomes a prefactor in the $\boldsymbol{C}$ matrix and can therefore be determined analytically. It turns out to be of order 1, so it does not change the noise parameter much.
The hyperparameters ℓ and k_0 can be determined using the Bayesian approach described in Chapter 19. The prefactor k_0 is for a given ℓ obtained analytically from Eq. 19.13, so the numerical optimization problem is only one-dimensional. The values $\ell = 1.67$ and $k_0 = 0.515$ are found using 500 data points. (This might vary significantly dependent on the particular set of data points).
After determination of the hyperparameters, the Gaussian process is trained on the same 500 data points.
The predictions E_{GP} for the heats of formation of the remaining 18428 perovskites can be seen in Figure 21.10 plotted versus the real DFT values E_{DFT}. The mean absolute error is 0.29 eV, while the root mean square error is 0.41 eV.
The distribution of errors, $\Delta E = E_{GP} - E_{DFT}$, are shown in Figure 21.11 (left). To the right the distribution of the ratio between the error of the prediction ΔE and the predicted error $\sqrt{\sigma^2(E)}$ is shown together with a Gaussian curve of width one. If the error prediction is correct the distribution should be close to the Gaussian.

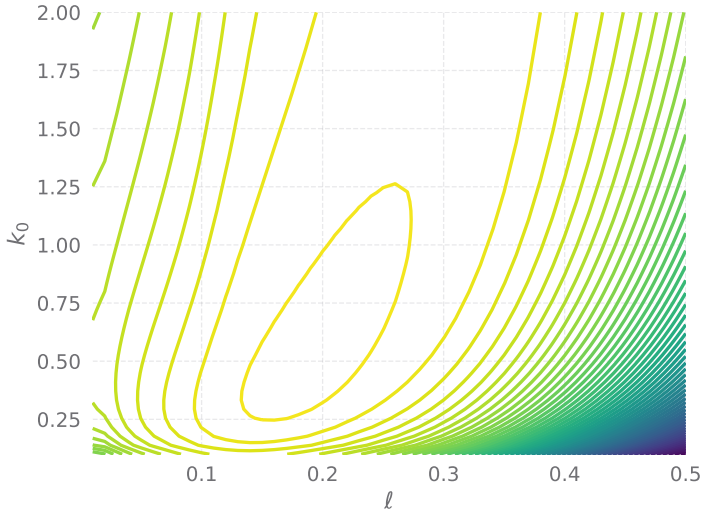

Figure 21.9: The log-likelihood for the hyperparameters ℓ and k_0.

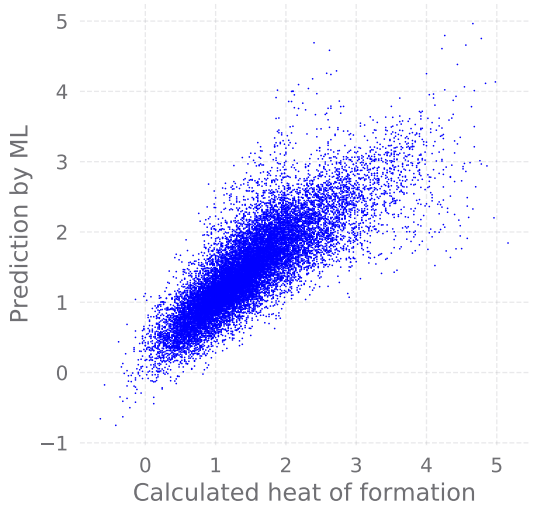

Figure 21.10: Comparison of predicted and real heats of formation for 18428 perovskites.

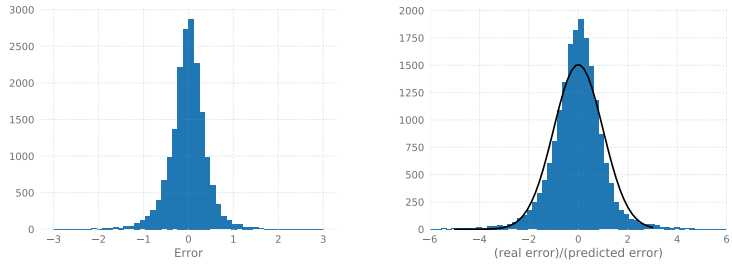

Figure 21.11: Left: The distribution of errors $E_{\mathrm{GP}} - E_{\mathrm{DFT}}$ on the heat of formation. Right: The distribution of the error in the prediction relative to the predicted one $(E_{\mathrm{GP}} - E_{\mathrm{DFT}})/\sigma(E_{\mathrm{GP}})$. The curve shows a Gaussian with width one.

Bibliography

[Jay03] E. T. Jaynes. *Probability theory: The logic of science*. Cambridge: Cambridge University Press, 2003.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[RW06] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. English. The MIT Press, 2006. ISBN: 9780262182539. URL: http://books.google.it/books?id=vWtwQgAACAAJ&dq=intitle:Gaussian+Processes+for+Machine+Learning&hl=&cd=1&source=gbs_api.

[SS06] D. S. Sivia and J. Skilling. *Data Analysis - A Bayesian Tutorial*. 2nd. Oxford Science Publications. Oxford University Press, 2006.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction; 2nd ed.* Springer Series in Statistics. Dordrecht: Springer, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7. URL: http://cds.cern.ch/record/1315326.

[Cas+12a] Ivano Eligio Castelli et al. "Computational screening of perovskite metal oxides for optimal solar light capture". English. In: *Energy & Environmental Science* 5.2 (2012), pp. 5814–5819. DOI: 10.1039/c1ee02717d. URL: http://xlink.rsc.org/?DOI=c1ee02717d.

[Cas+12b] Ivano Eligio Castelli et al. "New cubic perovskites for one- and two-photon water splitting using the computational materials repository". English. In: *Energy & Environmental Science* 5.10 (2012), p. 9034. DOI: 10.1039/c2ee22341d. URL: http://xlink.rsc.org/?DOI=c2ee22341d.

[Gha15] Zoubin Ghahramani. "Probabilistic machine learning and artificial intelligence". In: *Nature* 521.7553 (May 2015), pp. 452–459. DOI: 10.1038/nature14541. URL: http://www.nature.com/doifinder/10.1038/nature14541.