7 Estimation

In the first few chapters we examined parameterized regression models that we fit to data, often through the minimization of an objective function. In this chapter we consider a related problem, namely, how to determine the distributions of random variables based on measurements. The general problem is one of **inference**, which describes the probabilities of unknown parameters. Given a model, these probabilities can be derived using Bayes' Rule. The simplest use of these probabilities is to perform **estimation**, in which we attempt to come up with single "best" estimates of the unknown parameters. The relation between inference and regression is become more clear in this and subsequent chapters.

7.1 Learning a Binomial Distribution

For a simple example, let's return to coin-flipping. We flip a coin N times, with the result of the i-th flip denoted by a variable $\mathbf{c}_i$, where "$\mathbf{c}_i = \mathsf{heads}$" means that the i-th flip came up heads. The probability that the coin lands heads on any given trial is given by a parameter θ. We have no prior knowledge about the value of θ, and so our prior distribution on θ is uniform.[1] In other words, we describe θ as coming from a uniform distribution from 0 to 1, so $p(\theta) = 1$. So we believe that all values of θ are equally likely a priori (if we have not seen any data). We further assume that the individual coin flips are independent, i.e., $P(\mathbf{c}_{1:N}|\theta) = \prod_i p(\mathbf{c}_i|\theta)$. (The notation "$\mathbf{c}_{1:N}$" indicates the sequence of observations $\{\mathbf{c}_1, ..., \mathbf{c}_N\}$.) We summarize this model as follows:

Model: Coin-Flipping		
θ	$\sim$	$\mathcal{U}(0,1)$
$P(\mathbf{c} = \mathsf{heads})$	$=$	θ
$P(\mathbf{c}_{1:N}\|\theta)$	$=$	$\prod_i p(\mathbf{c}_i\|\theta)$

(1)

Suppose we wish to learn about a coin by flipping it 1000 times and observing the results $\mathbf{c}_{1:1000}$. Let's say the coin landed heads 750 times? What is our belief about θ, given this data, i.e., what is $p(\theta|\mathbf{c}_{1:1000})$? To find the answer we rely on the basic rules of probability, beginning with the Product Rule:

$$P(\mathbf{c}_{1:1000}, \theta) \;=\; P(\mathbf{c}_{1:1000}|\theta)\, p(\theta) \;=\; p(\theta|\mathbf{c}_{1:1000})\, P(\mathbf{c}_{1:1000}) \;. \tag{2}$$

Solving for the desired quantity gives:

$$p(\theta|\mathbf{c}_{1:1000}) \;\;=\;\; \frac{P(\mathbf{c}_{1:1000}|\theta)\, p(\theta)}{P(\mathbf{c}_{1:1000})} \;. \tag{3}$$

Because the flips are independent, and the coin landed heads on 750 of the flips, we can write the numerator as

$$P(\mathbf{c}_{1:1000}|\theta)\, p(\theta) \;=\; \prod_i P(\mathbf{c}_i|\theta) \;=\; \theta^{750}(1-\theta)^{1000-750} \;. \tag{4}$$

[1] We would usually expect a coin to be fair, i.e., the prior distribution for θ is peaked near 0.5.

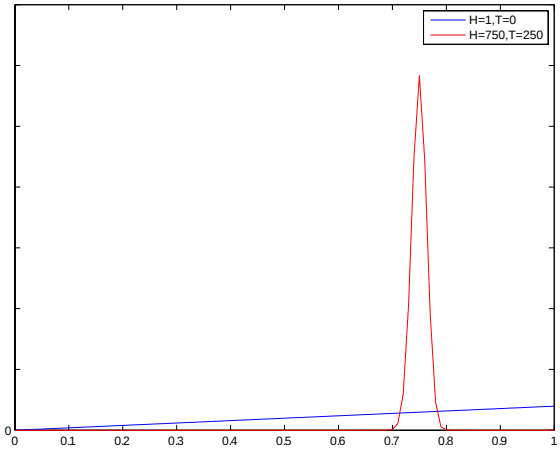

Figure 1: Posterior probability of θ from two different experiments: one with a single coin flip (landing heads), and 1000 coin flips (750 of which land heads). Note that the latter distribution is much more peaked.

The denominator may be solved for through marginalization, i.e.,

$$P(\mathbf{c}_{1:1000}) \;=\; \int_0^1 P(\mathbf{c}_{1:1000}, \theta)\, d\theta \;=\; \int_0^1 \theta^{750}(1-\theta)^{1000-750}\, d\theta \;=\; Z\,. \tag{5}$$

Here, Z is a constant (evaluating it requires more advanced math, but this is not necessary for our purposes here). The final probability distribution is given by

$$p(\theta|\mathbf{c}_{1:1000}) \;=\; \frac{1}{Z}\,\theta^{750}(1-\theta)^{1000-750}\,, \tag{6}$$

which is plotted in Figure 1. This probability distribution over θ quantifies our belief about θ *after* we've flipped the coin 1000 times.

Suppose we just take the peak of this distribution. From the graph, it can be seen that the peak is at $\theta = .75$. This makes sense; if a coin lands heads 75% of the time, then we would probably estimate that it will land heads 75% of the time in the future. More generally, suppose the coin lands heads H times out of N flips. We can compute the peak of the distribution as follows

$$\arg\max_\theta p(\theta|\mathbf{c}_{1:N}) \;=\; \frac{H}{N}\,. \tag{7}$$

(Deriving this is a good exercise to do on your own. Hint: minimize the negative log of $p(\theta|\mathbf{c}_{1:N})$).

7.2 Bayes' Rule

In general, given a model of the world in question, specified in terms of some unknown parameters, our goal is to determine the model from observed data. (In the coin-flip example, the model comprised the likelihood of the coin landing heads, and the prior over θ, while the data comprised

the results of the N coin flips.) We describe the probability model as $p(\mathsf{data}|\mathsf{model})$. If we knew model, then this tells us what data we should expect. Furthermore, we must have some prior beliefs as to what model is ($p(\mathsf{model})$), even if these beliefs are completely non-committal (e.g., a uniform distribution). Then, given data, what do we know about model?

Applying the product rule yields

$$p(\mathsf{data}, \mathsf{model}) \;=\; p(\mathsf{data}|\mathsf{model})\, p(\mathsf{model}) \;=\; p(\mathsf{model}|\mathsf{data})\, p(\mathsf{data}) \tag{8}$$

Solving for the desired distribution, gives a seemingly simple but powerful result, known widely as **Bayes' Rule:**

Bayes' Rule:

$$p(\mathsf{model}|\mathsf{data}) \;=\; \frac{p(\mathsf{data}|\mathsf{model})\, p(\mathsf{model})}{p(\mathsf{data})}$$

The different terms in Bayes' Rule are used so often that they all have names:

$$\underbrace{p(\mathsf{model}|\mathsf{data})}_{\text{posterior}} \;=\; \frac{\overbrace{P(\mathsf{data}|\mathsf{model})}^{\text{likelihood}}\overbrace{p(\mathsf{model})}^{\text{prior}}}{\underbrace{p(\mathsf{data})}_{\text{evidence}}} \tag{9}$$

- The **likelihood** distribution describes the likelihood of the observations data assuming the model is correct — it reflects our assumptions about how the data $\mathbf{c}$ were generated. With discrete data we express the likelihood as probability (as is the case with coin flips). For continuous data the likelihood is a density function.
- The **prior distribution** describes our assumptions about the model model without having observed any data data.
- The **posterior distribution** describes our knowledge of the model model based on both the data and the prior.
- The **evidence** is useful in model selection, and will be discussed later. Here, its only role is to normalize the posterior PDF so that it integrates to unity.

7.3 Parameter Estimation

In many situations we are mainly interested in finding a single estimate of the value of an unknown parameter. This is called **estimation**; i.e., determining the values of some unknown variables from observed data. In this chapter, we outline the problem, and describe some of the main approaches, including Maximum A Posteriori (MAP), and Maximum Likelihood (ML). Estimation is the most common form of learning. Given some data from the world, we wish to "learn" how the world behaves, which we describe in terms of a parameterized model.

Strictly speaking, parameter estimation is not justified by Bayesian probability theory, and can lead to a number of problems, such as overfitting and nonsensical results in extreme cases. Nonetheless, it is widely used in many problems. In later chapters we'll also see how we to determine a measure of confidence in estimated parameters.

7.3.1 MAP, ML and Bayes' Estimates

The MAP learning rule: choose the parameter value θ that maximizes the posterior distribution, i.e.,

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\ &= \arg\max_{\theta} P(\mathcal{D}|\theta)p(\theta) \ . \end{aligned} \tag{10}$$

Note that we don't need to be able to evaluate the evidence term $p(\mathcal{D})$ for MAP learning, since it does not depend on θ and may therefore be treated as a constant for MAP estimation.

Very often we will assume no prior assumptions about the value of θ, which we express as a **uniform prior**, where $p(\theta)$ assumed to be a uniform distribution over some suitably large range of values. In this case, $p(\theta)$ is constant, and can therefore also be ignored from MAP learning. Hence we are only maximizing the likelihood. Hence, the **Maximum Likelihood** (ML) learning principle is

$$\hat{\theta}_{ML} = \arg\max_{\theta} P(\mathcal{D}|\theta) \ . \tag{11}$$

It often turns out that it is more convenient to minimize the negative-log of the objective function. Because "$-\ln$" is a monotonic decreasing function, we can pose MAP estimation as:

$$\begin{aligned} \hat{\theta}_{MAP} &= \arg\max_{\theta} P(\mathcal{D}|\theta)p(\theta) \\ &= \arg\min_{\theta} -\ln\left(P(\mathcal{D}|\theta)p(\theta)\right) \\ &= \arg\min_{\theta} -\ln P(\mathcal{D}|\theta) - \ln p(\theta) \ . \end{aligned} \tag{12}$$

We can see that the objective conveniently breaks into two terms, one corresponding to the log likelihood and one corresponding to the log prior.

Both MAP and ML estimation essentially ignore uncertainty in the parameters. With thtese approaches we are choosing to put all our faith in the most probable model. This sometimes has surprising and undesirable consequences. For example, in the coin tossing example above, if one were to flip a coin just once and see a head, then the estimator in Eqn. (7) would tell us that the probability of the outcome being heads is 1. Sometimes a more suitable estimator is the expected value of the posterior distribution, rather than its maximum. This is called the **Bayes' estimate**.

For the coin tossing case above, one can show that the expected value of θ, under the posterior provides an estimate of the probability that is biased toward 1/2. That is,

$$\int_0^1 p(\theta|\mathbf{c}_{1:N})\,\theta\,d\theta = \frac{H+1}{N+2} \tag{13}$$

You can see that this value is always somewhat biased toward 1/2, but converges to the MAP estimate as N increases. Interestingly, even when there are is no data whatsoever, in which case the MAP estimate is undefined, the Bayes' estimate is simply 1/2.

7.4 Learning Gaussians

We now consider the problem of learning a Gaussian distribution from N training samples $\mathbf{x}_{1:N}$. Maximum likelihood learning of the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ entails maximizing the likelihood:

$$p(\mathbf{x}_{1:N}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{14}$$

We assume here that the data points come from a Gaussian. We further assume that they are drawn independently. We can therefore write the joint likelihood over the entire set of data as the product of the likelihoods for each individual datum, i.e.,

$$\begin{aligned} p(\mathbf{x}_{1:N}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) , \end{aligned} \tag{15}$$

where d is the dimensionality of the data vectors $\mathbf{x}_i$.

As mentioned above, it is often more convenient to minimize the negative log-likelihood:

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &\equiv -\ln p(\mathbf{x}_{1:N}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= -\sum_i \ln p(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \sum_i \frac{(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})}{2} + \frac{N}{2} \ln |\boldsymbol{\Sigma}| + \frac{Nd}{2} \ln(2\pi) . \end{aligned} \tag{16}$$

Solving for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by setting $\partial L/\partial \boldsymbol{\mu} = 0$ and $\partial L/\partial \boldsymbol{\Sigma} = 0$ (subject to the constraint that $\boldsymbol{\Sigma}$ is symmetric) gives the maximum likelihood estimates[2]:

$$\boldsymbol{\mu}^* = \frac{1}{N} \sum_i \mathbf{x}_i \tag{17}$$

$$\boldsymbol{\Sigma}^* = \frac{1}{N} \sum_i (\mathbf{x}_i - \boldsymbol{\mu}^*)(\mathbf{x}_i - \boldsymbol{\mu}^*)^T \tag{18}$$

The ML estimates make intuitive sense; we estimate the Gaussian's mean to be the sample mean of the data, and the Gaussian's covariance to be the sample covariance of the data. Maximum likelihood estimates usually make sense intuitively. This is very helpful when debugging the math; you can sometimes find bugs in derivations simply because the ML estimates do not look right.

7.5 MAP Nonlinear Regression

Let us revisit the nonlinear regression model from Section 3.1, but now admitting that there exists noise in measurements and modelling errors. We'll now write the generative model as

$$y = \mathbf{w}^T \mathbf{b}(\mathbf{x}) + \eta , \tag{19}$$

[2]Warning: the calculation for the optimal covariance matrix involves Lagrange multipliers and is not easy.

where η is a Gaussian random variable, i.e.,

$$\eta \sim \mathcal{N}(0, \sigma^2) \; . \tag{20}$$

We add this random variable to the regression equation in (19) to represent the fact that most models and most measurements involve some degree of error. We'll refer to this error as *noise*.

It is straightforward to show from basic probability theory that Eqn. (19) implies that, given $\mathbf{x}$ and $\mathbf{w}$, y is also Gaussian (i.e., has a Gaussian density). Formally,

$$p(y \,|\, \mathbf{x}, \mathbf{w}) \;=\; G(y;\, \mathbf{w}^T \mathbf{b}(\mathbf{x}),\, \sigma^2) \;\equiv\; \frac{1}{\sqrt{2\pi}\sigma}\, e^{-(y - \mathbf{w}^T \mathbf{b}(\mathbf{x}))^2 / 2\sigma^2} \tag{21}$$

(G is defined in the previous chapter.) It follows that, for a collection of N independent training data points, $(y_{1:N}, \mathbf{x}_{1:N})$, the likelihood is given by

$$\begin{aligned} p(y_{1:N} \,|\, \mathbf{w}, \mathbf{x}_{1:N}) \;&=\; \prod_{i=1}^{N} G(y_i;\, \mathbf{w}^T \mathbf{b}(\mathbf{x}_i),\, \sigma^2) \\ &=\; \frac{1}{(2\pi\sigma^2)^{N/2}}\, \exp\left(- \sum_{i=1}^{N} \frac{(y_i - \mathbf{w}^T \mathbf{b}(\mathbf{x}_i))^2}{2\sigma^2} \right) \end{aligned} \tag{22}$$

Furthermore, let's assume the following (weight decay) prior for the unknown weights $\mathbf{w}$:

$$\mathbf{w} \sim \mathcal{N}(0, \alpha \mathbf{I}) \; . \tag{23}$$

That is, for $\mathbf{w} \in \mathbb{R}^d$,

$$p(\mathbf{w}) \;=\; \prod_{k=1}^{d} \frac{1}{\sqrt{2\pi\alpha}}\, e^{-w_k^2 / 2\alpha} \;=\; \frac{1}{(2\pi\alpha)^{d/2}}\, e^{-\mathbf{w}^T \mathbf{w} / 2\alpha} \; . \tag{24}$$

Now, to estimate the model parameters (i.e., $\mathbf{w}$), let's consider the posterior distribution over $\mathbf{w}$ conditioned on our N training pairs, $(\mathbf{x}_i, y_i)$. Based on the formulation above, assuming independent training samples, it follows that

$$\begin{aligned} p(\mathbf{w} | y_{1:N}, \mathbf{x}_{1:N}) \;&=\; \frac{p(y_{1:N} | \mathbf{w}, \mathbf{x}_{1:N}) p(\mathbf{w} | \mathbf{x}_{1:N})}{p(y_{1:N} | \mathbf{x}_{1:N})} \\ &=\; \frac{\left(\prod_i p(y_i | \mathbf{w}, \mathbf{x}_i)\right) p(\mathbf{w})}{p(y_{1:N} | \mathbf{x}_{1:N})} \; . \end{aligned} \tag{25}$$

Note that $p(\mathbf{w} | \mathbf{x}_{1:N}) = p(\mathbf{w})$, since we can assume that $\mathbf{x}$ alone provides no information about $\mathbf{w}$.

In MAP estimation, we want to find the parameters $\mathbf{w}$ that maximize their posterior probability:

$$\begin{aligned} \mathbf{w}^* \;&=\; \arg\max_{\mathbf{w}} p(\mathbf{w} | y_{1:N}, \mathbf{x}_{1:N}) \\ &=\; \arg\min_{\mathbf{w}} - \ln p(\mathbf{w} | y_{1:N}, \mathbf{x}_{1:N}) \end{aligned} \tag{26}$$

The negative log-posterior is:

$$\begin{aligned} L(\mathbf{w}) &= -\ln p(\mathbf{w}|y_{1:N}, \mathbf{x}_{1:N}) \\ &= \left(\sum_i \frac{1}{2\sigma^2} (y_i - \mathbf{w}^T \mathbf{b}(\mathbf{x}_i))^2 \right) + \frac{N}{2} \ln(2\pi\sigma^2) \\ &\quad + \frac{1}{2\alpha} ||\mathbf{w}||^2 + \frac{d}{2} \ln(2\pi\alpha) + \ln p(y_{1:N}|\mathbf{x}_{1:N}) \end{aligned} \tag{27}$$

Now, we can discard terms that do not depend on $\mathbf{w}$, since they are irrelevant for optimization:

$$L(\mathbf{w}) = \left(\sum_i \frac{1}{2\sigma^2} (y_i - \mathbf{w}^T \mathbf{b}(\mathbf{x}_i))^2 \right) + \frac{1}{2\alpha} ||\mathbf{w}||^2 + \text{constants} \tag{28}$$

Furthermore, we can multiply by a constant, without changing where the optima are, so let us multiply the whole expression by $2\sigma^2$. Then, if we define $\lambda = \sigma^2/\alpha$, we have the exact same objective function as used in nonlinear regression with regularization. Hence, nonlinear least-squares with regularization is a form of MAP estimation, and can be optimized the same way. When the measurements are very reliable, then σ is small and we give the regularizer less influence on the estimate. But when the data are relatively noisy, so σ is larger, then the regularizer has more influence.

