CHAPTER 2

Estimating Probabilities

Many machine learning methods depend on probabilistic approaches. The reason is simple: when we are interested in learning some target function $f : X \rightarrow Y$, we can more generally learn the probabilistic function $P(Y|X)$. By using a probabilistic approach, we can design algorithms that learn functions with uncertain outcomes (e.g., predicting tomorrow's stock price) and that incorporate prior knowledge to guide learning (e.g., a bias that tomorrow's stock price is likely to be similar to today's price). This chapter describes joint probability distributions over many variables, and shows how they can be used to calculate a target $P(Y|X)$. It also considers the problem of learning, or estimating, probability distributions from training data, presenting the two most common approaches: maximum likelihood estimation and maximum a posteriori estimation.

1 Joint Probability Distributions

The key to building probabilistic models is to define a set of random variables, and to consider the joint probability distribution over them. For example, Table 1 defines a joint probability distribution over three random variables: a person's

Gender	HoursWorked	Wealth	probability
female	< 40.5	poor	0.2531
female	< 40.5	rich	0.0246
female	≥ 40.5	poor	0.0422
female	≥ 40.5	rich	0.0116
male	< 40.5	poor	0.3313
male	< 40.5	rich	0.0972
male	≥ 40.5	poor	0.1341
male	≥ 40.5	rich	0.1059

Table 1: **A Joint Probability Distribution.** This table defines a joint probability distribution over three random variables: Gender, HoursWorked, and Wealth.

Gender, the number of HoursWorked each week, and their Wealth. In general, defining a joint probability distribution over a set of discrete-valued variables involves three simple steps:

1. Define the random variables, and the set of values each variable can take on. For example, in Table 1 the variable *Gender* can take on the value *male* or *female*, the variable *HoursWorked* can take on the value "< 40.5' or "≥ 40.5," and *Wealth* can take on values *rich* or *poor*.

2. Create a table containing one row for each possible joint assignment of values to the variables. For example, Table 1 has 8 rows, corresponding to the 8 possible ways of jointly assigning values to three boolean-valued variables. More generally, if we have n boolean valued variables, there will be 2^n rows in the table.

3. Define a probability for each possible joint assignment of values to the variables. Because the rows cover every possible joint assignment of values, their probabilities must sum to 1.

The joint probability distribution is central to probabilistic inference, because once we know the joint distribution we can answer every possible probabilistic question that can be asked about these variables. We can calculate conditional or joint probabilities over *any* subset of the variables, given their joint distribution. This is accomplished by operating on the probabilities for the relevant rows in the table. For example, we can calculate:

- The probability that any single variable will take on any specific value. For example, we can calculate that the probability $P(\text{Gender} = \text{male}) = 0.6685$ for the joint distribution in Table 1, by summing the four rows for which Gender = male. Similarly, we can calculate the probability $P(\text{Wealth} = \text{rich}) = 0.2393$ by adding together the probabilities for the four rows covering the cases for which *Wealth=rich*.

- The probability that any subset of the variables will take on a particular joint assignment. For example, we can calculate that the probability $P(\text{Wealth=rich} \wedge \text{Gender=female}) = 0.0362$, by summing the two table rows that satisfy this joint assignment.

- Any conditional probability defined over subsets of the variables. Recall the definition of conditional probability $P(Y|X) = P(X \wedge Y)/P(X)$. We can calculate both the numerator and denominator in this definition by summing appropriate rows, to obtain the conditional probability. For example, according to Table 1, $P(\text{Wealth=rich}|\text{Gender=female}) = 0.0362/0.3315 = 0.1092$.

To summarize, if we know the joint probability distribution over an arbitrary set of random variables $\{X_1 \ldots X_n\}$, then we can calculate the conditional and joint probability distributions for arbitrary subsets of these variables (e.g., $P(X_n|X_1 \ldots X_{n-1})$). In theory, we can in this way solve any classification, regression, or other function approximation problem defined over these variables, and furthermore produce probabilistic rather than deterministic predictions for any given input to the target function.[1] For example, if we wish to learn to predict which people are rich or poor based on their gender and hours worked, we can use the above approach to simply calculate the probability distribution $P(\text{Wealth} \mid \text{Gender, HoursWorked})$.

1.1 Learning the Joint Distribution

How can we learn joint distributions from observed training data? In the example of Table 1 it will be easy if we begin with a large database containing, say, descriptions of a million people in terms of their values for our three variables. Given a large data set such as this, one can easily estimate a probability for each row in the table by calculating the fraction of database entries (people) that satisfy the joint assignment specified for that row. If thousands of database entries fall into each row, we will obtain highly reliable probability estimates using this strategy.

In other cases, however, it can be difficult to learn the joint distribution due to the very large amount of training data required. To see the point, consider how our learning problem would change if we were to add additional variables to describe a total of 100 boolean features for each person in Table 1 (e.g., we could add "do they have a college degree?", "are they healthy?"). Given 100 boolean features, the number of rows in the table would now expand to 2^{100}, which is greater than 10^{30}. Unfortunately, even if our database describes every single person on earth we would not have enough data to obtain reliable probability estimates for most rows. There are only approximately 10^{10} people on earth, which means that for most of the 10^{30} rows in our table, we would have zero training examples! This

[1] Of course if our random variables have continuous values instead of discrete, we would need an infinitely large table. In such cases we represent the joint distribution by a function instead of a table, but the principles for using the joint distribution remain unchanged.

is a significant problem given that real-world machine learning applications often use many more than 100 features to describe each example – for example, many learning algorithms for text analysis use millions of features to describe text in a given document.

To successfully address the issue of learning probabilities from available training data, we must (1) be smart about how we estimate probability parameters from available data, and (2) be smart about how we represent joint probability distributions.

2 Estimating Probabilities

Let us begin our discussion of how to estimate probabilities with a simple example, and explore two intuitive algorithms. It will turn out that these two intuitive algorithms illustrate the two primary approaches used in nearly all probabilistic machine learning algorithms.

In this simple example you have a coin, represented by the random variable X. If you flip this coin, it may turn up heads (indicated by $X = 1$) or tails ($X = 0$). The learning task is to estimate the probability that it will turn up heads; that is, to estimate $P(X=1)$. We will use θ to refer to the true (but unknown) probability of heads (e.g., $P(X=1) = \theta$), and use $\hat{\theta}$ to refer to our learned estimate of this true θ. You gather training data by flipping the coin n times, and observe that it turns up heads α_1 times, and tails α_0 times. Of course $n = \alpha_1 + \alpha_0$.

What is the most intuitive approach to estimating $\theta = P(X=1)$ from this training data? Most people immediately answer that we should estimate the probability by the fraction of flips that result in heads:

> **Probability estimation Algorithm 1 (maximum likelihood)**. Given observed training data producing α_1 total "heads," and α_0 total "tails," output the estimate
>
> $$\hat{\theta} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

For example, if we flip the coin 50 times, observing 24 heads and 26 tails, then we will estimate the probability $P(X = 1)$ to be $\hat{\theta} = 0.48$.

This approach is quite reasonable, and very intuitive. It is a good approach when we have plenty of training data. However, notice that if the training data is very scarce it can produce unreliable estimates. For example, if we observe only 3 flips of the coin, we might observe $\alpha_1 = 1$ and $\alpha_0 = 2$, producing the estimate $\hat{\theta} = 0.33$. How would we respond to this? If we have prior knowledge about the coin – for example, if we recognize it as a government minted coin which is likely to have θ close to 0.5 – then we might respond by still believing the probability is closer to 0.5 than to the algorithm 1 estimate $\hat{\theta} = 0.33$. This leads to our second intuitive algorithm: an algorithm that enables us to incorporate prior assumptions along with observed training data to produce our final estimate. In particular, Algorithm 2 allows us to express our prior assumptions or knowledge about the

coin by adding in any number of *imaginary* coin flips resulting in heads or tails. We can use this option of introducing γ_1 imaginary heads, and γ_0 imaginary tails, to express our prior assumptions:

> **Probability estimation Algorithm 2. (maximum a posteriori probability)**. Given observed training data producing α_1 observed "heads," and α_0 observed "tails," plus prior information expressed by introducing γ_1 imaginary "heads" and γ_0 imaginary "tails," output the estimate
>
> $$\hat{\theta} = \frac{(\alpha_1 + \gamma_1)}{(\alpha_1 + \gamma_1) + (\alpha_0 + \gamma_0)}$$

Note that Algorithm 2, like Algorithm 1, produces an estimate based on the proportion of coin flips that result in "heads." The only difference is that Algorithm 2 allows including optional imaginary flips that represent our prior assumptions about θ, in addition to actual observed data. Algorithm 2 has several attractive properties:

- It is easy to incorporate our prior assumptions about the *value* of θ by adjusting the *ratio* of γ_1 to γ_0. For example, if we have reason to assume that $\theta = 0.7$ we can add in $\gamma_1 = 7$ imaginary flips with $X = 1$, and $\gamma_0 = 3$ imaginary flips with $X = 0$.

- It is easy to express our *degree of certainty* about our prior knowledge, by adjusting the total *volume* of imaginary coin flips. For example, if we are highly certain of our prior belief that $\theta = 0.7$, then we might use priors of $\gamma_1 = 700$ and $\gamma_0 = 300$ instead of $\gamma_1 = 7$ and $\gamma_0 = 3$. By increasing the volume of imaginary examples, we effectively require a greater volume of contradictory observed data in order to produce a final estimate far from our prior assumed value.

- If we set $\gamma_1 = \gamma_0 = 0$, then Algorithm 2 produces exactly the same estimate as Algorithm 1. Algorithm 1 is just a special case of Algorithm 2.

- Asymptotically, as the volume of actual observed data grows toward infinity, the influence of our imaginary data goes to zero (the fixed number of imaginary coin flips becomes insignificant compared to a sufficiently large number of actual observations). In other words, Algorithm 2 behaves so that priors have the strongest influence when observations are scarce, and their influence gradually reduces as observations become more plentiful.

Both Algorithm 1 and Algorithm 2 are intuitively quite compelling. In fact, *these two algorithms exemplify the two most widely used approaches to machine learning of probabilistic models from training data.* They can be shown to follow from two different underlying principles. Algorithm 1 follows a principle called *Maximum Likelihood Estimation (MLE)*, in which we seek an estimate of θ that

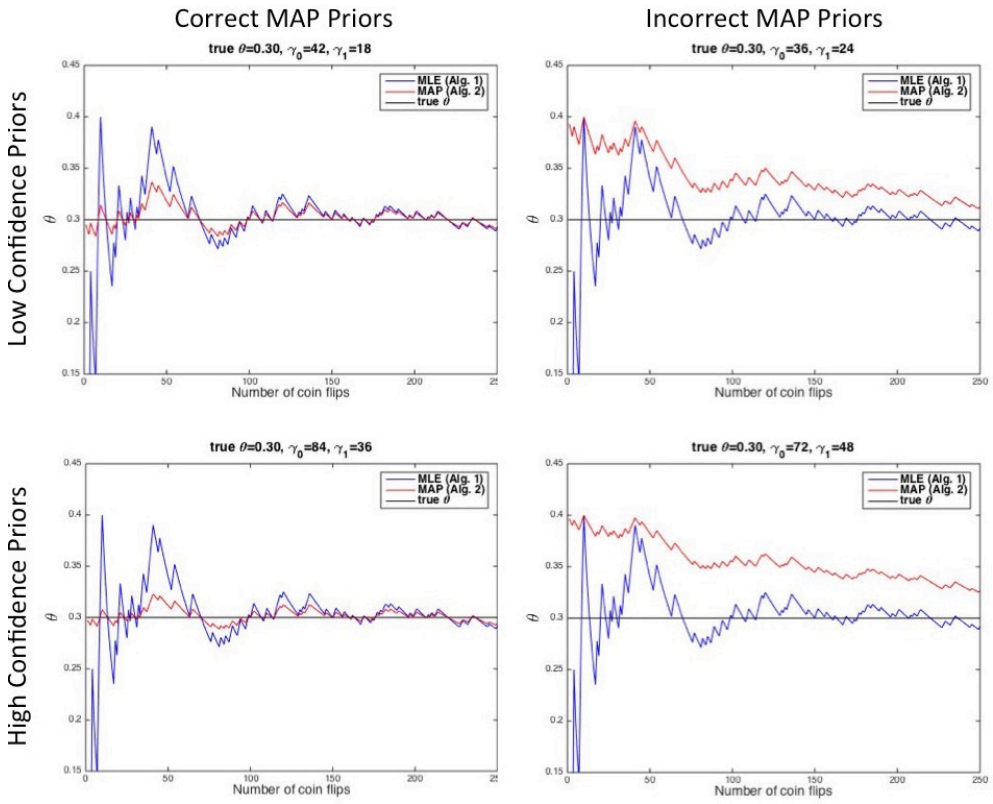

Figure 1: **MLE and MAP estimates of θ as the number of coin flips grows.** Data was generated by a random number generator that output a value of 1 with probability $\theta = 0.3$, and a value of 0 with probability of $(1-\theta) = 0.7$. Each plot shows the two estimates of θ as the number of observed coin flips grows. Plots on the left correspond to values of γ_1 and γ_0 that reflect the correct prior assumption about the value of θ, plots on the right reflect the incorrect prior assumption that θ is most probably 0.4. Plots in the top row reflect lower confidence in the prior assumption, by including only $60 = \gamma_1 + \gamma_0$ imaginary data points, whereas bottom plots assume 120. Note as the size of the data grows, the MLE and MAP estimates converge toward each other, and toward the correct estimate for θ.

---

maximizes the probability of the observed data. In fact we can prove (and will, below) that Algorithm 1 outputs an estimate of θ that makes the observed data at least as probable as any other possible estimate of θ. Algorithm 2 follows a different principle called *Maximum a Posteriori (MAP)* estimation, in which we seek the *estimate of* θ *that is itself most probable, given the observed data, plus background assumptions about its value.* Thus, the difference between these two principles is that Algorithm 2 assumes background knowledge is available, whereas Algorithm 1 does not. Both principles have been widely used to derive and to justify a vast range of machine learning algorithms, from Bayesian networks, to linear regression, to neural network learning. Our coin flip example represents just one of many such learning problems.

The experimental behavior of these two algorithms is shown in Figure 1. Here

the learning task is to estimate the unknown value of $\theta = P(X = 1)$ for a boolean-valued random variable X, based on a sample of n values of X drawn independently (e.g., n independent flips of a coin with probability θ of heads). In this figure, the true value of θ is 0.3, and the same sequence of training examples is used in each plot. Consider first the plot in the upper left. The blue line shows the estimates of θ produced by Algorithm 1 (MLE) as the number n of training examples grows. The red line shows the estimates produced by Algorithm 2, using the same training examples and using priors $\gamma_0 = 42$ and $\gamma_1 = 18$. This prior assumption aligns with the correct value of θ (i.e., $[\gamma_1/(\gamma_1 + \gamma_0)] = 0.3$). Note that as the number of training example coin flips grows, both algorithms converge toward the correct estimate of θ, though Algorithm 2 provides much better estimates than Algorithm 1 when little data is available. The bottom left plot shows the estimates if Algorithm 2 uses even more confident priors, captured by twice as many imaginary examples ($\gamma_0 = 84$ and $\gamma_1 = 36$). The two plots on the right side of the figure show the estimates produced when Algorithm 2 (MAP) uses incorrect priors (where $[\gamma_1/(\gamma_1 + \gamma_0)] = 0.4$). The difference between the top right and bottom right plots is again only a difference in the number of imaginary examples, reflecting the difference in confidence that θ should be close to 0.4.

2.1 Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation, often abbreviated MLE, estimates one or more probability parameters θ based on the principle that if we observe training data D, we should choose the value of θ that makes D most probable. When applied to the coin flipping problem discussed above, it yields Algorithm 1. The definition of the MLE in general is

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(D|\theta) \tag{1}$$

The intuition underlying this principle is simple: we are more likely to observe data D if we are in a world where the appearance of this data is highly probable. Therefore, we should estimate θ by assigning it whatever value maximizes the probability of having observed D.

Beginning with this principle for choosing among possible estimates of θ, it is possible to mathematically derive a formula for the value of θ that provably maximizes $P(D|\theta)$. Many machine learning algorithms are defined so that they provably learn a collection of parameter values that follow this maximum likelihood principle. Below we derive Algorithm 1 for our above coin flip example, beginning with the maximum likelihood principle.

To precisely define our coin flipping example, let X be a random variable which can take on either value 1 or 0, and let $\theta = P(X = 1)$ refer to the true, but possibly unknown, probability that a random draw of X will take on the value 1.[2] Assume we flip the coin X a number of times to produce training data D, in which

[2] A random variable defined in this way is called a Bernoulli random variable, and the probability distribution it follows, defined by θ, is called a Bernoulli distribution.

we observe $X=1$ a total of α_1 times, and $X=0$ a total of α_0 times. We further assume that the outcomes of the flips are independent (i.e., the result of one coin flip has no influence on other coin flips), and identically distributed (i.e., the same value of θ governs each coin flip). Taken together, these assumptions are that the coin flips are independent, identically distributed (which is often abbreviated to "i.i.d.").

The maximum likelihood principle involves choosing θ to maximize $P(D|\theta)$. Therefore, we must begin by writing an expression for $P(D|\theta)$, or equivalently $P(\alpha_1,\alpha_0|\theta)$ in terms of θ, then find an algorithm that chooses a value for θ that maximizes this quantity. To begin, note that if data D consists of just one coin flip, then $P(D|\theta)=\theta$ if that one coin flip results in $X=1$, and $P(D|\theta)=(1-\theta)$ if the result is instead $X=0$. Furthermore, if we observe a set of i.i.d. coin flips such as $D=\{1,1,0,1,0\}$, then we can easily calculate $P(D|\theta)$ by multiplying together the probabilities of each individual coin flip:

$$P(D=\{1,1,0,1,0\} \mid \theta) = \theta\cdot\theta\cdot(1-\theta)\cdot\theta\cdot(1-\theta) = \theta^3\cdot(1-\theta)^2$$

In other words, if we summarize D by the total number of observed times α_1 when $X=1$ and the number of times α_0 that $X=0$, we have in general

$$P(D=\langle\alpha_1,\alpha_0\rangle|\theta) = \theta^{\alpha_1}\ (1-\theta)^{\alpha_0} \tag{2}$$

The above expression gives us a formula for $P(D=\langle\alpha_1,\alpha_0\rangle|\theta)$. The quantity $P(D|\theta)$ is often called the *data likelihood*, or the *data likelihood function* because it expresses the probability of the observed data D as a function of θ. This likelihood function is often written $L(\theta)=P(D|\theta)$.

Our final step in this derivation is to determine the value of θ that maximizes the data likelihood function $P(D=\langle\alpha_1,\alpha_0\rangle|\theta)$. Notice that maximizing $P(D|\theta)$ with respect to θ is equivalent to maximizing its logarithm, $\ln P(D|\theta)$ with respect to θ, because $\ln(x)$ increases monotonically with x:

$$\arg\max_\theta P(D|\theta) \ = \ \arg\max_\theta \ln P(D|\theta)$$

It often simplifies the mathematics to maximize $\ln P(D|\theta)$ rather than $P(D|\theta)$, as is the case in our current example. In fact, this log likelihood is so common that it has its own notation, $\ell(\theta)=\ln P(D|\theta)$.

To find the value of θ that maximizes $\ln P(D|\theta)$, and therefore also maximizes $P(D|\theta)$, we can calculate the derivative of $\ln P(D=\langle\alpha_1,\alpha_0\rangle|\theta)$ with respect to θ, then solve for the value of θ that makes this derivative equal to zero. Because $\ln P(D|\theta)$ is a concave function of θ, the value of θ where this derivative is zero will be the value that maximizes $\ln P(D|\theta)$. First, we calculate the derivative of the log of the likelihood function of Eq. (2):

$$\begin{aligned}\frac{\partial\ell(\theta)}{\partial\theta} &= \frac{\partial \ln P(D|\theta)}{\partial\theta} \\ &= \frac{\partial\ \ln[\theta^{\alpha_1}(1-\theta)^{\alpha_0}]}{\partial\theta}\end{aligned}$$

$$\begin{aligned}
&= \frac{\partial\left[\alpha_1 \ln\theta + \alpha_0 \ln(1-\theta)\right]}{\partial\theta} \\
&= \alpha_1 \frac{\partial \ln\theta}{\partial\theta} + \alpha_0 \frac{\partial \ln(1-\theta)}{\partial\theta} \\
&= \alpha_1 \frac{\partial \ln\theta}{\partial\theta} + \alpha_0 \frac{\partial \ln(1-\theta)}{\partial(1-\theta)} \cdot \frac{\partial(1-\theta)}{\partial\theta} \\
\frac{\partial \ell(\theta)}{\partial\theta} &= \alpha_1 \frac{1}{\theta} + \alpha_0 \frac{1}{(1-\theta)} \cdot (-1) \qquad (3)
\end{aligned}$$

where the last step follows from the equality $\frac{\partial \ln x}{\partial x} = \frac{1}{x}$, and where the next to last step follows from the chain rule $\frac{\partial f(x)}{\partial x} = \frac{\partial f(x)}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}$.

Finally, to calculate the value of θ that maximizes $\ell(\theta)$, we set the derivative in equation (3) to zero, and solve for θ.

$$\begin{aligned}
0 &= \alpha_1 \frac{1}{\theta} - \alpha_0 \frac{1}{1-\theta} \\
\alpha_0 \frac{1}{1-\theta} &= \alpha_1 \frac{1}{\theta} \\
\alpha_0 \theta &= \alpha_1 (1-\theta) \\
(\alpha_1 + \alpha_0)\theta &= \alpha_1 \\
\theta &= \frac{\alpha_1}{\alpha_1 + \alpha_0} \qquad (4)
\end{aligned}$$

Thus, we have derived in equation (4) the intuitive Algorithm 1 for estimating θ, starting from the principle that we want to choose the value of θ that maximizes $P(D|\theta)$.

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(D|\theta) = \arg\max_{\theta} \ln P(D|\theta) = \frac{\alpha_1}{\alpha_1 + \alpha_0} \qquad (5)$$

This same maximum likelihood principle is used as the basis for deriving many machine learning algorithms for more complex problems where the solution is not so intuitively obvious.

2.2 Maximum a Posteriori Probability Estimation (MAP)

Maximum a Posteriori Estimation, often abbreviated to MAP estimation, estimates one or more probability parameters θ based on the principle that we should choose the value of θ that is most probable, given the observed data D and our prior assumptions summarized by $P(\theta)$.

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\theta|D)$$

When applied to the coin flipping problem discussed above, it yields Algorithm 2. Using Bayes rule, we can rewrite the MAP principle as:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\theta|D) = \arg\max_{\theta} \frac{P(D|\theta)P(\theta)}{P(D)}$$

and given that $P(D)$ does not depend on θ, we can simplify this by ignoring the denominator:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\theta|D) = \arg\max_{\theta} P(D|\theta)P(\theta) \tag{6}$$

Comparing this to the MLE principle described in equation (1), we see that whereas the MLE principle is to choose θ to maximize $P(D|\theta)$, the MAP principle instead maximizes $P(D|\theta)P(\theta)$. The only difference is the extra $P(\theta)$.

To produce a MAP estimate for θ we must specify a prior distribution $P(\theta)$ that summarizes our a priori assumptions about the value of θ. In the case where data is generated by multiple i.i.d. draws of a Bernoulli random variable, as in our coin flip example, the most common form of prior is a Beta distribution:

$$P(\theta) = Beta(\beta_0, \beta_1) = \frac{\theta^{\beta_1 - 1}\ (1-\theta)^{\beta_0 - 1}}{B(\beta_0, \beta_1)} \tag{7}$$

Here β_0 and β_1 are parameters whose values we must specify in advance to define a specific $P(\theta)$. As we shall see, choosing values for β_0 and β_1 corresponds to choosing the number of imaginary examples γ_0 and γ_1 in the above Algorithm 2. The denominator $B(\beta_0, \beta_1)$ is a normalization term defined by the function B, which assures the probability integrates to one, but which is independent of θ.

As defined in Eq. (6), the MAP estimate involves choosing the value of θ that maximizes $P(D|\theta)P(\theta)$. Recall we already have an expression for $P(D|\theta)$ in Eq. (2). Combining this with the above expression for $P(\theta)$ we have:

$$\begin{aligned}
\hat{\theta}^{MAP} &= \arg\max_{\theta} P(D|\theta)P(\theta) \\
&= \arg\max_{\theta}\ \theta^{\alpha_1}(1-\theta)^{\alpha_0}\ \ \frac{\theta^{\beta_1 - 1}\ (1-\theta)^{\beta_0 - 1}}{B(\beta_0, \beta_1)} \\
&= \arg\max_{\theta}\ \frac{\theta^{\alpha_1 + \beta_1 - 1}\ (1-\theta)^{\alpha_0 + \beta_0 - 1}}{B(\beta_0, \beta_1)} \\
&= \arg\max_{\theta}\ \theta^{\alpha_1 + \beta_1 - 1}\ (1-\theta)^{\alpha_0 + \beta_0 - 1}
\end{aligned} \tag{8}$$

where the final line follows from the previous line because $B(\beta_0, \beta_1)$ is independent of θ.

How can we solve for the value of θ that maximizes the expression in Eq. (8)? Fortunately, we have already answered this question! Notice that the quantity we seek to maximize in Eq. (8) can be made identical to the likelihood function in Eq. (2) if we substitute $(\alpha_1 + \beta_1 - 1)$ for α_1 in Eq. (2), and substitute $(\alpha_0 + \beta_0 - 1)$ for α_0. We can therefore reuse the derivation of $\hat{\theta}^{MLE}$ beginning from Eq. (2) and ending with Eq. (4), simply by carrying through this substitution. Applying this same substitution to Eq. (4) implies the solution to Eq. (8) is therefore

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(D|\theta)P(\theta) = \frac{(\alpha_1 + \beta_1 - 1)}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)} \tag{9}$$

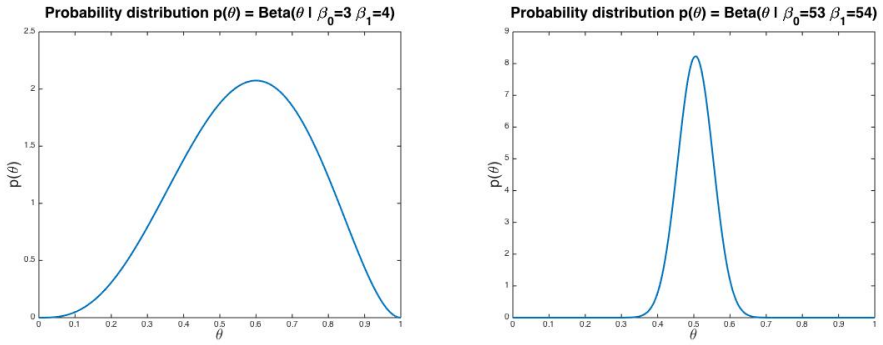

Figure 2: **Prior (left) and posterior (right) probability distributions on θ in a MAP estimate for θ.** Consider estimating $\theta = P(X = 1)$ for a boolean valued (Bernoulli) random variable X. Suppose we set a prior $P(\theta)$ defined by a Beta distribution with $\beta_0 = 3, \beta_1 = 4$, as shown on the left. Suppose further that we now observe data D consisting of 100 examples: 50 where we observe $X = 1$ and 50 where $X = 0$. Then the posterior probability $P(\theta|D)$ over θ given this observed data D, which is proportional to $P(D|\theta)P(\theta)$, is another Beta distribution with $\beta_0 = 53, \beta_1 = 54$, as shown on the right. Notice that both distributions assign non-zero probability to every possible value of θ between 0 and 1, though the posterior distribution has most of its probability mass near $\theta = 0.5$.

Thus, we have derived in Eq. (9) the intuitive Algorithm 2 for estimating θ, starting from the principle that we want to choose the value of θ that maximizes $P(\theta|D)$. The number γ_1 of imaginary "heads" in Algorithm 2 is equal to $\beta_1 - 1$, and the number γ_0 of imaginary "tails" is equal to $\beta_0 - 1$. This same maximum a posteriori probability principle is used as the basis for deriving many machine learning algorithms for more complex problems where the solution is not so intuitively obvious as it is in our coin flipping example.

2.2.1 MAP Priors and Posteriors

Why did we chose above to use the $Beta(\beta_0, \beta_1)$ family of probability distributions to define our prior probability $P(\theta)$ when calculating the MAP estimate $\hat{\theta}^{MAP}$? We made this choice because the Beta distribution has a functional form that is the same as the data likelihood $P(D|\theta)$ in our problem, so that when we multiply these two forms together to get the quantity $P(D|\theta)P(\theta)$, this product is easily expressed as yet another expression of this same functional form. Furthermore, in this product the β_0 and β_1 parameters that define our Beta distribution play exactly the same role as the observed data counts; that is, they capture the effect of the Beta prior $P(\theta)$ in a form that is equivalent to specifying the number of imaginary examples used in our earlier Algorithm 2[3].

Figure 2 shows an example of the prior $P(\theta)$ and posterior $P(\theta|D) \propto P(D|\theta)P(\theta)$ distributions corresponding to a MAP estimate in our example.

[3]More precisely, the number of imaginary examples λ_i in Algorithm 2 is given by $\beta_i - 1$.

The $Beta(\beta_0, \beta_1)$ distribution defined in Eq. (7) is called the **conjugate prior** for the binomial likelihood function $\theta^{\alpha_1}(1-\theta)^{\alpha_0}$, because the posterior distribution $P(D|\theta)P(\theta)$ is also a *Beta* distribution. More generally, any $P(\theta)$ is called the conjugate prior for a likelihood function $L(\theta) = P(D|\theta)$ if the posterior $P(\theta|D)$ is of the same form as $P(\theta)$.

3 Working with Other Probability Distributions

Formally, the probability distribution we considered above is called a Bernoulli distribution: it governs a random variable X which can take on two possible values, 0 or 1, and this Bernoulli distribution is defined by the single parameter θ (i.e., $P(X = 1) = \theta$, and $P(X = 0) = (1 - \theta)$. We will sometimes refer to a boolean-valued random variable which is governed by a Bernoulli distribution as a Bernoulli random variable. As noted above, the conjugate prior for estimating the parameter θ of a Bernoulli distribution is the Beta distribution.

3.1 Discrete Valued Random Variables with Many Values

If we have a random variable that can take on more than two values then we need more than one parameter to describe the probability distribution for that variable. For example, consider a six-sided die which, when rolled, can come up with any of 6 possible results.

A common approach to characterizing such n-valued random variables is to use a generalization of the Bernoulli distribution called a Categorical distribution, where we assign a different probability to each possible value of the random variable. For example, we might model a six-sided die as a random variable X that can take on the values 1 through 6, and represent its probability distribution by a vector $\boldsymbol{\theta}$ of six different parameters $\boldsymbol{\theta} = \langle \theta_1 \ldots \theta_6 \rangle$, where the parameter θ_i describes the probability that X will take on its i^{th} value; that is, $\theta_i = P(X = i)$. The likelihood function $L(\boldsymbol{\theta}) = P(D|\boldsymbol{\theta})$ for estimating the vector $\boldsymbol{\theta}$ of parameters from observed rolls of the die is a simple generalization of the likelihood for estimating a Bernoulli distribution. It takes the form of a product $L(\boldsymbol{\theta}) = P(D|\boldsymbol{\theta}) = \theta_1^{\alpha_1}\ \theta_2^{\alpha_2} \ldots \theta_n^{\alpha_n}$, where α_i indicates the observed count of times when the value $X = i$ was observed in the data. Given a sample of data drawn from a particular Categorical distribution for a random variable that can take on n possible values, the maximum likelihood estimate for θ_i is given by

$$\hat{\theta}_i^{MLE} = \frac{\alpha_i}{\alpha_1 + \ldots + \alpha_n} \tag{10}$$

where α_j indicates the number of times the value $X = j$ was observed in the data. Note that just like the case of a binary-valued random variable (see eq. 5), the MLE is simply the fraction of times the particular value was observed in the data.

If we prefer a MAP estimate for an n-valued random variable governed by a Categorical distribution, we use the conjugate prior for the Categorical distribution, which is called the Dirichlet distribution. Of course given that the Categorical

distribution has n different θ_i parameters, its prior will have to specify the probability for each joint assignment of these n parameters. The Dirichlet distribution is a generalization of the Beta distribution, and has the form

$$P(\theta_1, \ldots \theta_n) = \frac{\theta_1^{(\beta_1-1)}\theta_2^{(\beta_2-1)} \ldots \theta_n^{(\beta_n-1)}}{B(\beta_1, \ldots, \beta_n)}$$

where the denominator is again a normalizing function to assure that the total probability mass is 1, and where this normalizing function $B(\beta_1, \ldots, \beta_n)$ is independent of the vector of parameters $\boldsymbol{\theta} = \langle \theta_1 \ldots \theta_n \rangle$ and therefore can be ignored when deriving their MAP estimates.

The MAP estimate for each θ_i for a Categorial distribution is given by

$$\hat{\theta}_i^{MAP} = \frac{(\alpha_i + \beta_i - 1)}{(\alpha_1 + \beta_1 - 1) + \ldots + (\alpha_n + \beta_n - 1)} \tag{11}$$

where α_j indicates the number of times the value $X = j$ was observed in the data, and where the β_js are the parameters of the Dirichlet prior which reflects our prior knowledge or assumptions. Here again, we can view the MAP estimate as combining the observed data given by the α_j values with $\beta_j - 1$ additional imaginary observations for $X = j$. Comparing this formula to the earlier formula giving the MAP estimate for a Bernoulli random variable (eq. 9), it is easy to see that this is a direct generalization of that simpler case, and that it again follows the intuition of our earlier Algorithm 2.

4 What You Should Know

The main points of this chapter include:

- Joint probability distributions lie at the core of probabilistic machine learning approaches. Given the joint probability distribution $P(X_1 \ldots X_n)$ over a set of random variables, it is possible in principle to compute *any* joint or conditional probability defined over *any* subset of these variables.

- Learning, or estimating, the joint probability distribution from training data can be easy if the data set is large compared to the number of distinct probability terms we must estimate. But in many practical problems the data is more sparse, requiring methods that rely on prior knowledge or assumptions, in addition to observed data.

- *Maximum likelihood estimation* (MLE) is one of two widely used principles for estimating the parameters that define a probability distribution. This principle is to choose the set of parameter values $\hat{\theta}^{MLE}$ that makes the observed training data most probable (over all the possible choices of θ):

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(\text{data}|\theta)$$

In many cases, maximum likelihood estimates correspond to the intuitive notion that we should base probability estimates on observed ratios. For example, given the problem of estimating the probability that a coin will turn up heads, given α_1 observed flips resulting in heads, and α_0 observed flips resulting in tails, the maximum likelihood estimate corresponds exactly to taking the fraction of flips that turn up heads:

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(\text{data}|\theta) = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

- *Maximium a posteriori probability* (MAP) estimation is the other of the two widely used principles. This principle is to choose the most probable value of θ, given the observed training data plus a prior probability distribution $P(\theta)$ which captures prior knowledge or assumptions about the value of θ:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\theta|\text{data}) \; = \; \arg\max_{\theta} P(\text{data}|\theta)P(\theta)$$

In many cases, MAP estimates correspond to the intuitive notion that we can represent prior assumptions by making up "imaginary" data which reflects these assumptions. For example, the MAP estimate for the above coin flip example, assuming a prior $P(\theta) = Beta(\gamma_0 + 1, \gamma_1 + 1)$, yields a MAP estimate which is equivalent to the MLE estimate if we simply add in an imaginary γ_1 heads and γ_0 tails to the actual observed α_1 heads and α_0 tails:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\text{data}|\theta)P(\theta) = \frac{(\alpha_1 + \gamma_1)}{(\alpha_1 + \gamma_1) + (\alpha_0 + \gamma_0)}$$

EXERCISES

1. In the MAP estimation of θ for our Bernoulli random variable X in this chapter, we used a Beta(β_0, β_1) prior probability distribution to capture our prior beliefs about the prior probability of different values of θ, before seeing the observed data.

    - Plot this prior probability distribution over θ, corresponding to the number of imaginary examples used in the top left plot of Figure 1 (i.e., $\gamma_0 = 42, \gamma_1 = 18$). Specifically create a plot showing the prior probability (vertical axis) for each possible value of θ between 0 and 1 (horizontal axis), as represented by the prior distribution Beta(β_0, β_1). Recall the correspondence $\beta_i = \gamma_i + 1$. Note you will want to write a simple computer program to create this plot.
    - Above, you plotted the *prior* probability over possible values of θ. Now plot the *posterior* probability distribution over θ given that prior, plus observed data in which 6 heads ($X = 1$) were observed, along with 9 tails ($X = 0$).

- View the plot you created above to visually determine the approximate Maximum a Posterior probability estimate θ^{MAP}. What is it? What is the *exact* value of the MAP estimate? What is the exact value of the Maximum Likelihood Estimate θ^{MLE}?
- How do you think your plot of the posterior probability would change if you altered the Beta prior distribution to use $\gamma_0 = 420, \gamma_1 = 180$? (hint: it's ok to actually plot this). What if you changed the Beta prior to $\gamma_0 = 32, \gamma_1 = 28$?

5 Acknowledgements

I very much appreciate receiving helpful comments on earlier drafts of this chapter from Ondřej Filip, Ayush Garg, Akshay Mishra and Tao Chen. Andrew Moore provided the data summary shown in Table 1.

REFERENCES

Mitchell, T (1997). *Machine Learning*, McGraw Hill.

Wasserman, L. (2004). *All of Statistics*, Springer-Verlag.