
CHAPTER 2

Estimating Probabilities

Machine Learning

Copyright © 2015. Tom M. Mitchell. All rights reserved.

DRAFT OF January 31, 2015

*PLEASE DO NOT DISTRIBUTE WITHOUT AUTHOR'S
PERMISSION*

This is a rough draft chapter intended for inclusion in the upcoming second edition of the textbook *Machine Learning*, T.M. Mitchell, McGraw Hill. You are welcome to use this for educational purposes, but do not duplicate or repost it on the internet. For online copies of this and other materials related to this book, visit the web site www.cs.cmu.edu/~tom/mlbook.html. Please send suggestions for improvements, or suggested exercises, to Tom.Mitchell@cmu.edu.

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. We can calculate conditional or joint probabilities over *any* subset of 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 $P(\text{Wealth}=\text{rich} \wedge$

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 distribution over an arbitrary set of variables $\{X_1 \dots X_n\}$, then we can calculate the conditional and joint probability distributions for arbitrary subsets of these variables (e.g., $P(X_n|X_1 \dots X_{n-1})$). 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

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) who 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 is a significant problem given that real-world machine learning applications often use many more than 100 features to describe each example, and that learning such probability terms is central to probabilistic machine learning algorithms.

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.

¹Of 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.

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, 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 $\hat{\theta} = 0.48$.

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 for $X = 0$.
- 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 a gradually reduced influence as observations become more plentiful.

Both Algorithm 1 and Algorithm 2 are intuitively quite compelling. In fact, these two algorithms represent the two primary approaches to machine learning of probabilistic models from training data. They can be shown to follow from two different principles. Algorithm 1 follows a principle called *Maximum Likelihood Estimation*, in which we estimate θ by choosing the value which *maximizes the probability of the observed data, given θ* . Algorithm 2 follows a principle called *Maximum a Posteriori Probability* estimation, in which we estimate θ by choosing the value that *maximizes the value of θ given the observed data*. Both principles can be applied in a vast range of probabilistic learning problems. Our coin flip example represents just one of many such problems.

Below we derive Algorithms 1 and 2 directly from these two principles. It is very useful to understand these derivations because we will reuse the logic of these derivations over and over to derive a wide range of probabilistic machine learning algorithms, from Bayesian graphical models, to polynomial regression.

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

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

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 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 quantify. 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 $D = \langle 1, 1, 0, 1, 0 \rangle$, then we can easily calculate $P(D|\theta)$ by multiplying together the probabilities of each individual coin flip:

$$P(D = \langle 1, 1, 0, 1, 0 \rangle | \theta) = \theta \cdot \theta \cdot (1-\theta) \cdot \theta \cdot (1-\theta) = \theta^3 \cdot (1-\theta)^2$$

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

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

²A 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.

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 *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 derivation is to determine the value of θ that maximizes the 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 its argument 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. First, we calculate the derivative of the log of the likelihood function of Eq. (2):

$$\begin{aligned} \frac{\partial \ell(\theta)}{\partial \theta} &= \frac{\partial \ln P(D|\theta)}{\partial \theta} \\ &= \frac{\partial \ln [\theta^{\alpha_1} (1-\theta)^{\alpha_0}]}{\partial \theta} \\ &= \frac{\partial [\alpha_1 \ln \theta + \alpha_0 \ln(1-\theta)]}{\partial \theta} \\ &= \alpha_1 \frac{\partial \ln \theta}{\partial \theta} + \alpha_0 \frac{\partial \ln(1-\theta)}{\partial \theta} \\ &= \alpha_1 \frac{\partial \ln \theta}{\partial \theta} + \alpha_0 \frac{\partial \ln(1-\theta)}{\partial (1-\theta)} \cdot \frac{\partial (1-\theta)}{\partial \theta} \\ \frac{\partial \ell(\theta)}{\partial \theta} &= \alpha_1 \frac{1}{\theta} + \alpha_0 \frac{1}{(1-\theta)} \cdot (-1) \end{aligned} \tag{3}$$

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 .

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

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 $P(D|\theta)$.

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

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:

$$\hat{\theta}^{MAP} = \arg \max_{\theta} P(\theta|D) = \arg \max_{\theta} P(D|\theta)P(\theta) \quad (5)$$

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:

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

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_1 and γ_0 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. (5), 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:

$$\begin{aligned}
 \hat{\theta}^{MAP} &= \arg \max_{\theta} P(D|\theta)P(\theta) \\
 &= \arg \max_{\theta} \theta^{\alpha_1} (1-\theta)^{\alpha_0} \frac{\theta^{\beta_1-1} (1-\theta)^{\beta_0-1}}{B(\beta_0, \beta_1)} \\
 &= \arg \max_{\theta} \frac{\theta^{\alpha_1+\beta_1-1} (1-\theta)^{\alpha_0+\beta_0-1}}{B(\beta_0, \beta_1)} \\
 &= \arg \max_{\theta} \theta^{\alpha_1+\beta_1-1} (1-\theta)^{\alpha_0+\beta_0-1} \tag{6}
 \end{aligned}$$

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. (6)? Fortunately, we have already answered this question! Notice that the quantity we seek to maximize in Eq. (6) can be made identical to the likelihood function in Eq. (2) if we substitute $\alpha_1 + \beta_1 - 1$ for α 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. (6) is therefore

$$\hat{\theta}^{MAP} = \arg \max_{\theta} P(D|\theta)P(\theta) = \frac{(\alpha_1 + \beta_1 - 1)}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)} \tag{7}$$

Thus, we have derived in Eq. (7) 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..

3 Notes on Terminology

A boolean valued random variable $X \in \{0, 1\}$, governed by the probability distribution $P(X=1) = \theta$; $P(X=0) = (1-\theta)$ is called a **Bernoulli random variable**, and this probability distribution is called a **Bernoulli distribution**. A convenient mathematical expression for a Bernoulli distribution $P(X)$ is:

$$P(X=x) = \theta^x \cdot (1-\theta)^{1-x}$$

The $Beta(\beta_0, \beta_1)$ distribution 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 $P(\theta)$.

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 $P(X_1 \dots X_n)$ 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 $\hat{\theta}^{MLE}$ which 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 our intuition that we should simply take the fraction of flips that turn up heads:

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

- *Maximum 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 $P(\theta) = \text{Beta}(\gamma_0 + 1, \gamma_1 + 1)$, 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:

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

REFERENCES

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

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