3 *Generative models for discrete data*

3.1 Introduction

In Section 2.2.3.2, we discussed how to classify a feature vector $\mathbf{x}$ by applying Bayes rule to a generative classifier of the form

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x}|y = c, \boldsymbol{\theta})p(y = c|\boldsymbol{\theta}) \tag{3.1}$$

The key to using such models is specifying a suitable form for the class-conditional density $p(\mathbf{x}|y = c, \boldsymbol{\theta})$, which defines what kind of data we expect to see in each class. In this chapter, we focus on the case where the observed data are discrete symbols. We also discuss how to infer the unknown parameters $\boldsymbol{\theta}$ of such models.

3.2 Bayesian concept learning

Consider how a child learns to understand the meaning of a word, such as "dog". Presumably the child's parents point out positive examples of this concept, saying such things as, "look at the cute dog!", or "mind the doggy", etc. However, it is very unlikely that they provide negative examples, by saying "look at that non-dog". Certainly, negative examples may be obtained during an active learning process — the child says "look at the dog" and the parent says "that's a cat, dear, not a dog" — but psychological research has shown that people can learn concepts from positive examples alone (Xu and Tenenbaum 2007).

We can think of learning the meaning of a word as equivalent to **concept learning**, which in turn is equivalent to binary classification. To see this, define $f(x) = 1$ if x is an example of the concept C, and $f(x) = 0$ otherwise. Then the goal is to learn the indicator function f, which just defines which elements are in the set C. By allowing for uncertainty about the definition of f, or equivalently the elements of C, we can emulate **fuzzy set theory**, but using standard probability calculus. Note that standard binary classification techniques require positive and negative examples. By contrast, we will devise a way to learn from positive examples alone.

For pedagogical purposes, we will consider a very simple example of concept learning called the **number game**, based on part of Josh Tenenbaum's PhD thesis (Tenenbaum 1999). The game proceeds as follows. I choose some simple arithmetical concept C, such as "prime number" or "a number between 1 and 10". I then give you a series of randomly chosen positive examples $\mathcal{D} = \{x_1, \ldots, x_N\}$ drawn from C, and ask you whether some new test case $\tilde{x}$ belongs to C, i.e., I ask you to classify $\tilde{x}$.

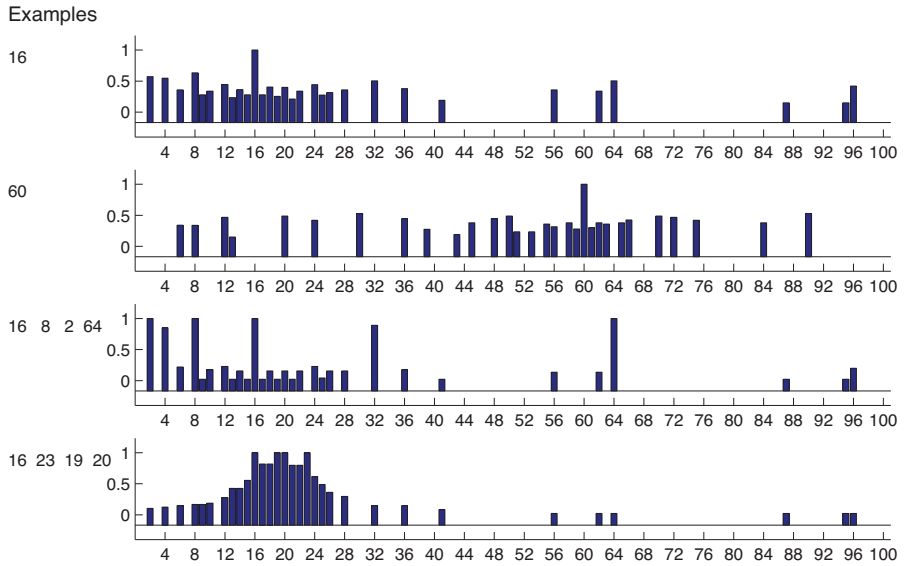


Figure 3.1 Empirical predictive distribution averaged over 8 humans in the number game. First two rows: after seeing $\mathcal{D} = \{16\}$ and $\mathcal{D} = \{60\}$. This illustrates diffuse similarity. Third row: after seeing $\mathcal{D} = \{16, 8, 2, 64\}$. This illustrates rule-like behavior (powers of 2). Bottom row: after seeing $\mathcal{D} = \{16, 23, 19, 20\}$. This illustrates focussed similarity (numbers near 20). Source: Figure 5.5 of (Tenenbaum 1999). Used with kind permission of Josh Tenenbaum.

Suppose, for simplicity, that all numbers are integers between 1 and 100. Now suppose I tell you "16" is a positive example of the concept. What other numbers do you think are positive? 17? 6? 32? 99? It's hard to tell with only one example, so your predictions will be quite vague. Presumably numbers that are similar in some sense to 16 are more likely. But similar in what way? 17 is similar, because it is "close by", 6 is similar because it has a digit in common, 32 is similar because it is also even and a power of 2, but 99 does not seem similar. Thus some numbers are more likely than others. We can represent this as a probability distribution, $p(\tilde{x}|\mathcal{D})$, which is the probability that $\tilde{x} \in C$ given the data $\mathcal{D}$ for any $\tilde{x} \in \{1, \ldots, 100\}$. This is called the **posterior predictive distribution**. Figure 3.1(top) shows the predictive distribution of people derived from a lab experiment. We see that people predict numbers that are similar to 16, under a variety of kinds of similarity.

Now suppose I tell you that 8, 2 and 64 are *also* positive examples. Now you may guess that the hidden concept is "powers of two". This is an example of **induction**. Given this hypothesis, the predictive distribution is quite specific, and puts most of its mass on powers of 2, as shown in Figure 3.1(third row). If instead I tell you the data is $\mathcal{D} = \{16, 23, 19, 20\}$, you will get a different kind of **generalization gradient**, as shown in Figure 3.1(bottom).

How can we explain this behavior and emulate it in a machine? The classic approach to induction is to suppose we have a **hypothesis space** of concepts, $\mathcal{H}$, such as: odd numbers, even numbers, all numbers between 1 and 100, powers of two, all numbers ending in j (for

$0 \leq j \leq 9$), etc. The subset of $\mathcal{H}$ that is consistent with the data D is called the **version space**. As we see more examples, the version space shrinks and we become increasingly certain about the concept (Mitchell 1997).

However, the version space is not the whole story. After seeing $\mathcal{D} = \{16\}$, there are many consistent rules; how do you combine them to predict if $\tilde{x} \in C$? Also, after seeing $\mathcal{D} = \{16, 8, 2, 64\}$, why did you choose the rule "powers of two" and not, say, "all even numbers", or "powers of two except for 32", both of which are equally consistent with the evidence? We will now provide a Bayesian explanation for this.

3.2.1 Likelihood

We must explain why we chose $h_{\text{two}} \triangleq$"powers of two", and not, say, $h_{\text{even}} \triangleq$ "even numbers" after seeing $\mathcal{D} = \{16, 8, 2, 64\}$, given that both hypotheses are consistent with the evidence. The key intuition is that we want to avoid **suspicious coincidences**. If the true concept was even numbers, how come we only saw numbers that happened to be powers of two?

To formalize this, let us assume that examples are sampled uniformly at random from the **extension** of a concept. (The extension of a concept is just the set of numbers that belong to it, e.g., the extension of h_{even} is $\{2, 4, 6, \ldots, 98, 100\}$; the extension of "numbers ending in 9" is $\{9, 19, \ldots, 99\}$.) Tenenbaum calls this the **strong sampling assumption**. Given this assumption, the probability of independently sampling N items (with replacement) from h is given by

$$p(\mathcal{D}|h) = \left[\frac{1}{\text{size}(h)}\right]^N = \left[\frac{1}{|h|}\right]^N \tag{3.2}$$

This crucial equation embodies what Tenenbaum calls the **size principle**, which means the model favors the simplest (smallest) hypothesis consistent with the data. This is more commonly known as **Occam's razor**.[1]

To see how it works, let $\mathcal{D} = \{16\}$. Then $p(\mathcal{D}|h_{\text{two}}) = 1/6$, since there are only 6 powers of two less than 100, but $p(\mathcal{D}|h_{\text{even}}) = 1/50$, since there are 50 even numbers. So the likelihood that $h = h_{\text{two}}$ is higher than if $h = h_{\text{even}}$. After 4 examples, the likelihood of h_{two} is $(1/6)^4 = 7.7 \times 10^{-4}$, whereas the likelihood of h_{even} is $(1/50)^4 = 1.6 \times 10^{-7}$. This is a **likelihood ratio** of almost 5000:1 in favor of h_{two}. This quantifies our earlier intuition that $D = \{16, 8, 2, 64\}$ would be a very suspicious coincidence if generated by h_{even}.

3.2.2 Prior

Suppose $D = \{16, 8, 2, 64\}$. Given this data, the concept $h' =$"powers of two except 32" is more likely than $h =$"powers of two", since h' does not need to explain the coincidence that 32 is missing from the set of examples.

However, the hypothesis $h' =$"powers of two except 32" seems "conceptually unnatural". We can capture such intution by assigning low prior probability to unnatural concepts. Of course, your prior might be different than mine. This **subjective** aspect of Bayesian reasoning is a source of much controversy, since it means, for example, that a child and a math professor

1. William of Occam (also spelt Ockham) was an English monk and philosopher, 1288–1348.

will reach different answers. In fact, they presumably not only have different priors, but also different hypothesis spaces. However, we can finesse that by defining the hypothesis space of the child and the math professor to be the same, and then setting the child's prior weight to be zero on certain "advanced" concepts. Thus there is no sharp distinction between the prior and the hypothesis space.

Although the subjectivity of the prior is controversial, it is actually quite useful. If you are told the numbers are from some arithmetic rule, then given 1200, 1500, 900 and 1400, you may think 400 is likely but 1183 is unlikely. But if you are told that the numbers are examples of healthy cholesterol levels, you would probably think 400 is unlikely and 1183 is likely. Thus we see that the prior is the mechanism by which background knowledge can be brought to bear on a problem. Without this, rapid learning (i.e., from small samples sizes) is impossible.

So, what prior should we use? For illustration purposes, let us use a simple prior which puts uniform probability on 30 simple arithmetical concepts, such as "even numbers", "odd numbers", "prime numbers", "numbers ending in 9", etc. To make things more interesting, we make the concepts even and odd more likely apriori. We also include two "unnatural" concepts, namely "powers of 2, plus 37" and "powers of 2, except 32", but give them low prior weight. See Figure 3.2(a) for a plot of this prior. We will consider a slightly more sophisticated prior later on.

3.2.3 Posterior

The posterior is simply the likelihood times the prior, normalized. In this context we have

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{\sum_{h' \in \mathcal{H}} p(\mathcal{D}, h')} = \frac{p(h)\mathbb{I}(\mathcal{D} \in h)/|h|^N}{\sum_{h' \in \mathcal{H}} p(h')\mathbb{I}(\mathcal{D} \in h')/|h'|^N} \tag{3.3}$$

where $\mathbb{I}(\mathcal{D} \in h)$ is 1 **iff** (iff and only if) all the data are in the extension of the hypothesis h. Figure 3.2 plots the prior, likelihood and posterior after seeing $\mathcal{D} = \{16\}$. We see that the posterior is a combination of prior and likelihood. In the case of most of the concepts, the prior is uniform, so the posterior is proportional to the likelihood. However, the "unnatural" concepts of "powers of 2, plus 37" and "powers of 2, except 32" have low posterior support, despite having high likelihood, due to the low prior. Conversely, the concept of odd numbers has low posterior support, despite having a high prior, due to the low likelihood.

Figure 3.3 plots the prior, likelihood and posterior after seeing $\mathcal{D} = \{16, 8, 2, 64\}$. Now the likelihood is much more peaked on the powers of two concept, so this dominates the posterior. Essentially the learner has an **aha** moment, and figures out the true concept. (Here we see the need for the low prior on the unnatural concepts, otherwise we would have overfit the data and picked "powers of 2, except for 32".)

In general, when we have enough data, the posterior $p(h|\mathcal{D})$ becomes peaked on a single concept, namely the MAP estimate, i.e.,

$$p(h|\mathcal{D}) \to \delta_{\hat{h}^{MAP}}(h) \tag{3.4}$$

where $\hat{h}^{MAP} = \text{argmax}_h \, p(h|\mathcal{D})$ is the posterior mode, and where δ is the **Dirac measure** defined by

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \tag{3.5}$$

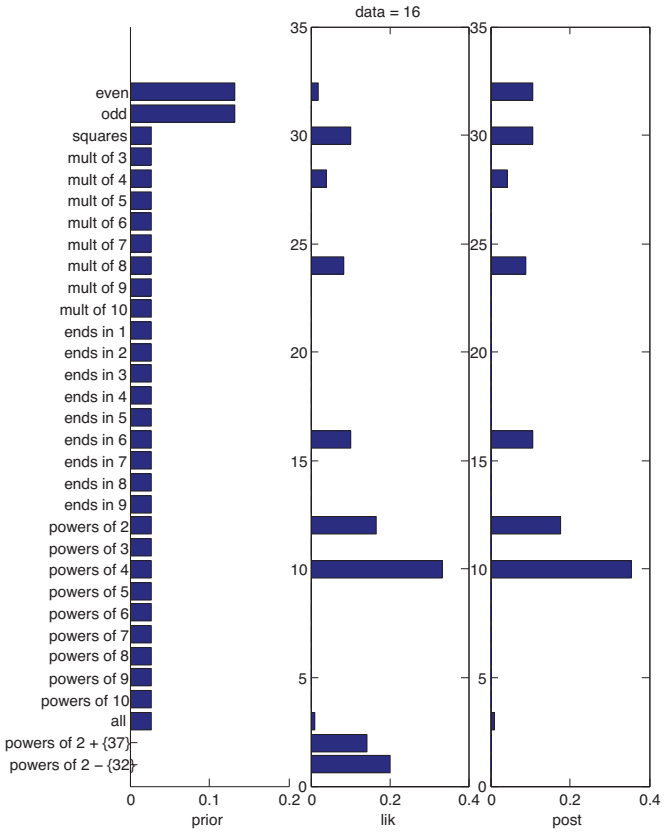


Figure 3.2 Prior, likelihood and posterior for $\mathcal{D} = \{16\}$. Based on (Tenenbaum 1999). Figure generated by `numbersGame`.

Note that the MAP estimate can be written as

$$\hat{h}^{MAP} = \underset{h}{\operatorname{argmax}}\, p(\mathcal{D}|h)p(h) = \underset{h}{\operatorname{argmax}}\, [\log p(\mathcal{D}|h) + \log p(h)] \tag{3.6}$$

Since the likelihood term depends exponentially on N, and the prior stays constant, as we get more and more data, the MAP estimate converges towards the **maximum likelihood estimate** or **MLE**:

$$\hat{h}^{mle} \triangleq \underset{h}{\operatorname{argmax}}\, p(\mathcal{D}|h) = \underset{h}{\operatorname{argmax}} \log p(\mathcal{D}|h) \tag{3.7}$$

In other words, if we have enough data, we see that the **data overwhelms the prior**. In this

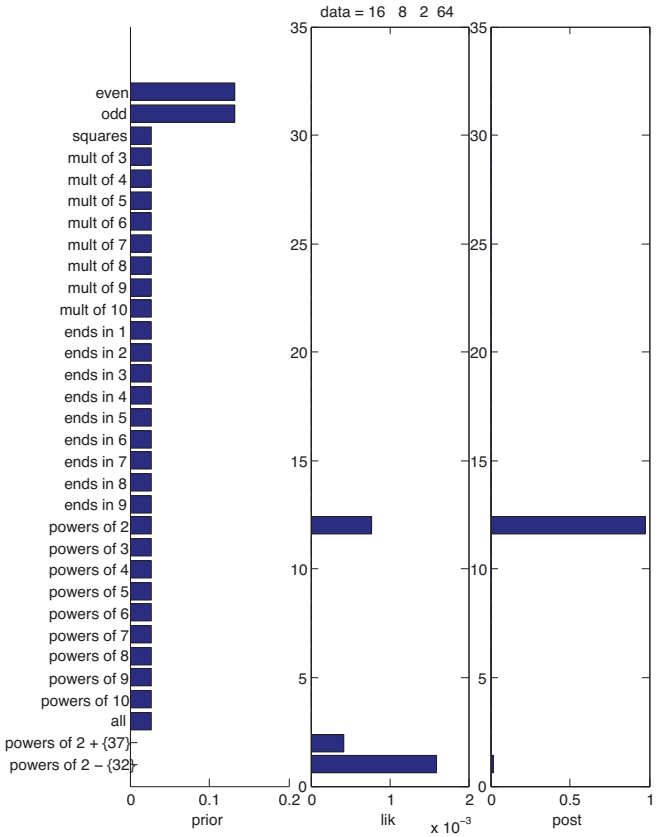


Figure 3.3 Prior, likelihood and posterior for $\mathcal{D} = \{16, 8, 2, 64\}$. Based on (Tenenbaum 1999). Figure generated by `numbersGame`.

case, the MAP estimate converges towards the MLE.

If the true hypothesis is in the hypothesis space, then the MAP/ ML estimate will converge upon this hypothesis. Thus we say that Bayesian inference (and ML estimation) are consistent estimators (see Section 6.4.1 for details). We also say that the hypothesis space is **identifiable in the limit**, meaning we can recover the truth in the limit of infinite data. If our hypothesis class is not rich enough to represent the "truth" (which will usually be the case), we will converge on the hypothesis that is as close as possible to the truth. However, formalizing this notion of "closeness" is beyond the scope of this chapter.

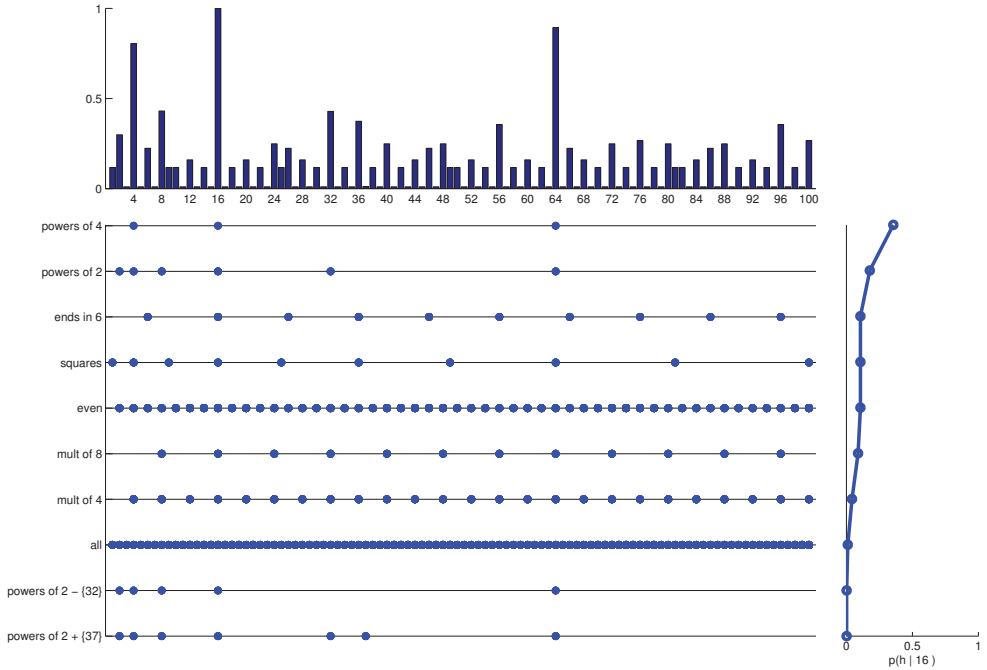


Figure 3.4 Posterior over hypotheses and the corresponding predictive distribution after seeing one example, $\mathcal{D} = \{16\}$. A dot means this number is consistent with this hypothesis. The graph $p(h|\mathcal{D})$ on the right is the weight given to hypothesis h. By taking a weighed sum of dots, we get $p(\tilde{x} \in C|\mathcal{D})$ (top). Based on Figure 2.9 of (Tenenbaum 1999). Figure generated by `numbersGame`.

3.2.4 Posterior predictive distribution

The posterior is our internal **belief state** about the world. The way to test if our beliefs are justified is to use them to predict objectively observable quantities (this is the basis of the scientific method). Specifically, the posterior predictive distribution in this context is given by

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(y = 1|\tilde{x}, h)p(h|\mathcal{D}) \tag{3.8}$$

This is just a weighted average of the predictions of each individual hypothesis and is called **Bayes model averaging** (Hoeting et al. 1999). This is illustrated in Figure 3.4. The dots at the bottom show the predictions from each hypothesis; the vertical curve on the right shows the weight associated with each hypothesis. If we multiply each row by its weight and add up, we get the distribution at the top.

When we have a small and/or ambiguous dataset, the posterior $p(h|\mathcal{D})$ is vague, which induces a broad predictive distribution. However, once we have "figured things out", the posterior becomes a delta function centered at the MAP estimate. In this case, the predictive distribution

becomes

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(\tilde{x}|h)\delta_{\hat{h}}(h) = p(\tilde{x}|\hat{h}) \tag{3.9}$$

This is called a **plug-in approximation** to the predictive density and is very widely used, due to its simplicity. However, in general, this under-represents our uncertainty, and our predictions will not be as "smooth" as when using BMA. We will see more examples of this later in the book.

Although MAP learning is simple, it cannot explain the gradual shift from similarity-based reasoning (with uncertain posteriors) to rule-based reasoning (with certain posteriors). For example, suppose we observe $\mathcal{D} = \{16\}$. If we use the simple prior above, the minimal consistent hypothesis is "all powers of 4", so only 4 and 16 get a non-zero probability of being predicted. This is of course an example of overfitting. Given $\mathcal{D} = \{16, 8, 2, 64\}$, the MAP hypothesis is "all powers of two". Thus the plug-in predictive distribution gets broader (or stays the same) as we see more data: it starts narrow, but is forced to broaden as it seems more data. In contrast, in the Bayesian approach, we start broad and then narrow down as we learn more, which makes more intuitive sense. In particular, given $\mathcal{D} = \{16\}$, there are many hypotheses with non-negligible posterior support, so the predictive distribution is broad. However, when we see $\mathcal{D} = \{16, 8, 2, 64\}$, the posterior concentrates its mass on one hypothesis, so the predictive distribution becomes narrower. So the predictions made by a plug-in approach and a Bayesian approach are quite different in the small sample regime, although they converge to the same answer as we see more data.

3.2.5 A more complex prior

To model human behavior, Tenenbaum used a slightly more sophisticated prior which was derived by analysing some experimental data of how people measure similarity between numbers; see (Tenenbaum 1999, p208) for details. The result is a set of arithmetical concepts similar to those mentioned above, plus all intervals between n and m for $1 \leq n, m \leq 100$. (Note that these hypotheses are not mutually exclusive.) Thus the prior is a **mixture** of two priors, one over arithmetical rules, and one over intervals:

$$p(h) = \pi_0 p_{\text{rules}}(h) + (1 - \pi_0) p_{\text{interval}}(h) \tag{3.10}$$

The only free parameter in the model is the relative weight, π_0, given to these two parts of the prior. The results are not very sensitive to this value, so long as $\pi_0 > 0.5$, reflecting the fact that people are more likely to think of concepts defined by rules. The predictive distribution of the model, using this larger hypothesis space, is shown in Figure 3.5. It is strikingly similar to the human predictive distribution, shown in Figure 3.1, even though it was not fit to human data (modulo the choice of hypothesis space).

3.3 The beta-binomial model

The number game involved inferring a distribution over a discrete variable drawn from a finite hypothesis space, $h \in \mathcal{H}$, given a series of discrete observations. This made the computations particularly simple: we just needed to sum, multiply and divide. However, in many applications, the unknown parameters are continuous, so the hypothesis space is (some subset) of $\mathbb{R}^K$, where

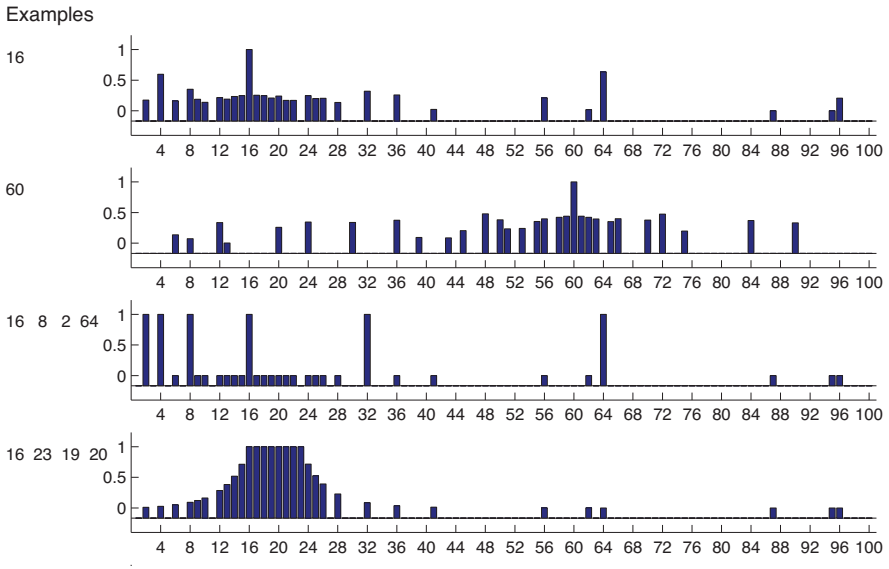


Figure 3.5 Predictive distributions for the model using the full hypothesis space. Compare to Figure 3.1. The predictions of the Bayesian model are only plotted for those values of $\tilde{x}$ for which human data is available; this is why the top line looks sparser than Figure 3.4. Source: Figure 5.6 of (Tenenbaum 1999). Used with kind permission of Josh Tenenbaum.

K is the number of parameters. This complicates the mathematics, since we have to replace sums with integrals. However, the basic ideas are the same.

We will illustrate this by considering the problem of inferring the probability that a coin shows up heads, given a series of observed coin tosses. Although this might seem trivial, it turns out that this model forms the basis of many of the methods we will consider later in this book, including naive Bayes classifiers, Markov models, etc. It is historically important, since it was the example which was analyzed in Bayes' original paper of 1763. (Bayes' analysis was subsequently generalized by Pierre-Simon Laplace, creating what we now call "Bayes rule" — see (Stigler 1986) for further historical details.)

We will follow our now-familiar recipe of specifying the likelihood and prior, and deriving the posterior and posterior predictive.

3.3.1 Likelihood

Suppose $X_i \sim \text{Ber}(\theta)$, where $X_i = 1$ represents "heads", $X_i = 0$ represents "tails", and $\theta \in [0, 1]$ is the rate parameter (probability of heads). If the data are iid, the likelihood has the form

$$p(\mathcal{D}|\theta) = \theta^{N_1}(1-\theta)^{N_0} \tag{3.11}$$

where we have $N_1 = \sum_{i=1}^{N} \mathbb{I}(x_i = 1)$ heads and $N_0 = \sum_{i=1}^{N} \mathbb{I}(x_i = 0)$ tails. These two counts are called the **sufficient statistics** of the data, since this is all we need to know about $\mathcal{D}$ to infer θ. (An alternative set of sufficient statistics are N_1 and $N = N_0 + N_1$.)

More formally, we say $\mathbf{s}(\mathcal{D})$ is a sufficient statistic for data $\mathcal{D}$ if $p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\theta}|\mathbf{s}(data))$. If we use a uniform prior, this is equivalent to saying $p(\mathcal{D}|\boldsymbol{\theta} \propto p(\mathbf{s}(\mathcal{D})|\boldsymbol{\theta})$. Consequently, if we have two datasets with the same sufficient statistics, we will infer the same value for $\boldsymbol{\theta}$.

Now suppose the data consists of the count of the number of heads N_1 observed in a fixed number $N = N_1 + N_0$ of trials. In this case, we have $N_1 \sim \text{Bin}(N, \theta)$, where Bin represents the binomial distribution, which has the following pmf:

$$\text{Bin}(k|n, \theta) \triangleq \binom{n}{k} \theta^k (1-\theta)^{n-k} \tag{3.12}$$

Since $\binom{n}{k}$ is a constant independent of θ, the likelihood for the binomial sampling model is the same as the likelihood for the Bernoulli model. So any inferences we make about θ will be the same whether we observe the counts, $\mathcal{D} = (N_1, N)$, or a sequence of trials, $\mathcal{D} = \{x_1, \ldots, x_N\}$.

3.3.2 Prior

We need a prior which has support over the interval $[0, 1]$. To make the math easier, it would convenient if the prior had the same form as the likelihood, i.e., if the prior looked like

$$p(\theta) \propto \theta^{\gamma_1} (1-\theta)^{\gamma_2} \tag{3.13}$$

for some prior parameters γ_1 and γ_2. If this were the case, then we could easily evaluate the posterior by simply adding up the exponents:

$$p(\theta) \propto p(\mathcal{D}|\theta) p(\theta) = \theta^{N_1} (1-\theta)^{N_0} \theta^{\gamma_1} (1-\theta)^{\gamma_2} = \theta^{N_1+\gamma_1} (1-\theta)^{N_0+\gamma_2} \tag{3.14}$$

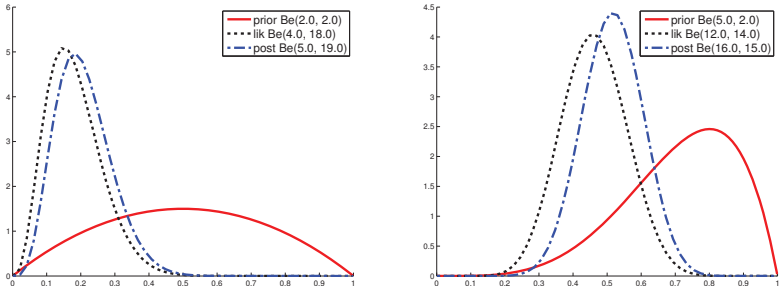
When the prior and the posterior have the same form, we say that the prior is a **conjugate prior** for the corresponding likelihood. Conjugate priors are widely used because they simplify computation, and are easy to interpret, as we see below.

In the case of the Bernoulli, the conjugate prior is the beta distribution, which we encountered in Section 2.4.5:

$$\text{Beta}(\theta|a, b) \propto \theta^{a-1} (1-\theta)^{b-1} \tag{3.15}$$

The parameters of the prior are called **hyper-parameters**. We can set them in order to encode our prior beliefs. For example, to encode our beliefs that θ has mean 0.7 and standard deviation 0.2, we set $a = 2.975$ and $b = 1.275$ (Exercise 3.15). Or to encode our beliefs that θ has mean 0.15 and that we think it lives in the interval $(0.05, 0.30)$ with probability, then we find $a = 4.5$ and $b = 25.5$ (Exercise 3.16).

If we know "nothing" about θ, except that it lies in the interval $[0, 1]$, we can use a uniform prior, which is a kind of uninformative prior (see Section 5.4.2 for details). The uniform distribution can be represented by a beta distribution with $a = b = 1$.



(a) (b)

Figure 3.6 (a) Updating a Beta(2, 2) prior with a Binomial likelihood with sufficient statistics $N_1 = 3, N_0 = 17$ to yield a Beta(5,19) posterior. (b) Updating a Beta(5, 2) prior with a Binomial likelihood with sufficient statistics $N_1 = 11, N_0 = 13$ to yield a Beta(16, 15) posterior. Figure generated by `binomialBetaPosteriorDemo`.

3.3.3 Posterior

If we multiply the likelihood by the beta prior we get the following posterior (following Equation 3.14):

$$p(\theta|\mathcal{D}) \propto \text{Bin}(N_1|\theta, N_0 + N_1)\text{Beta}(\theta|a, b)\text{Beta}(\theta|N_1 + a, N_0 + b) \tag{3.16}$$

In particular, the posterior is obtained by adding the prior hyper-parameters to the empirical counts. For this reason, the hyper-parameters are known as **pseudo counts**. The strength of the prior, also known as the **effective sample size** of the prior, is the sum of the pseudo counts, $a + b$; this plays a role analogous to the data set size, $N_1 + N_0 = N$.

Figure 3.6(a) gives an example where we update a weak Beta(2,2) prior with a peaked likelihood function, corresponding to a large sample size; we see that the posterior is essentially identical to the likelihood: since the data has overwhelmed the prior. Figure 3.6(b) gives an example where we update a strong Beta(5,2) prior with a peaked likelihood function; now we see that the posterior is a "compromise" between the prior and likelihood.

Note that updating the posterior sequentially is equivalent to updating in a single batch. To see this, suppose we have two data sets $\mathcal{D}_a$ and $\mathcal{D}_b$ with sufficient statistics N_1^a, N_0^a and N_1^b, N_0^b. Let $N_1 = N_1^a + N_1^b$ and $N_0 = N_0^a + N_0^b$ be the sufficient statistics of the combined datasets. In batch mode we have

$$p(\theta|\mathcal{D}_a, \mathcal{D}_b) \propto \text{Bin}(N_1|\theta, N_1 + N_0)\text{Beta}(\theta|a, b) \propto \text{Beta}(\theta|N_1 + a, N_0 + b) \tag{3.17}$$

In sequential mode, we have

$$\begin{aligned} p(\theta|\mathcal{D}_a, \mathcal{D}_b) &\propto p(\mathcal{D}_b|\theta)p(\theta|\mathcal{D}_a) && (3.18)\\ &\propto \text{Bin}(N_1^b|\theta, N_1^b + N_0^b)\text{Beta}(\theta|N_1^a + a, N_0^a + b) && (3.19)\\ &\propto \text{Beta}(\theta|\ N_1^a + N_1^b + a, N_0^a + N_0^b + b) && (3.20) \end{aligned}$$

This makes Bayesian inference particularly well-suited to **online learning**, as we will see later.

3.3.3.1 Posterior mean and mode

From Equation 2.62, the MAP estimate is given by

$$\hat{\theta}_{MAP} = \frac{a + N_1 - 1}{a + b + N - 2} \tag{3.21}$$

If we use a uniform prior, then the MAP estimate reduces to the MLE, which is just the empirical fraction of heads:

$$\hat{\theta}_{MLE} = \frac{N_1}{N} \tag{3.22}$$

This makes intuitive sense, but it can also be derived by applying elementary calculus to maximize the likelihood function in Equation 3.11. (Exercise 3.1).

By contrast, the posterior mean is given by,

$$\overline{\theta} = \frac{a + N_1}{a + b + N} \tag{3.23}$$

This difference between the mode and the mean will prove important later.

We will now show that the posterior mean is convex combination of the prior mean and the MLE, which captures the notion that the posterior is a compromise between what we previously believed and what the data is telling us.

Let $\alpha_0 = a + b$ be the **equivalent sample size** of the prior, which controls its strength, and let the prior mean be $m_1 = a/\alpha_0$. Then the posterior mean is given by

$$\mathbb{E}[\theta|\mathcal{D}] = \frac{\alpha_0 m_1 + N_1}{N + \alpha_0} = \frac{\alpha_0}{N + \alpha_0} m_1 + \frac{N}{N + \alpha_0} \frac{N_1}{N} = \lambda m_1 + (1 - \lambda)\hat{\theta}_{MLE} \tag{3.24}$$

where $\lambda = \frac{\alpha_0}{N + \alpha_0}$ is the ratio of the prior to posterior equivalent sample size. So the weaker the prior, the smaller is λ, and hence the closer the posterior mean is to the MLE. One can show similarly that the posterior mode is a convex combination of the prior mode and the MLE, and that it too converges to the MLE.

3.3.3.2 Posterior variance

The mean and mode are point estimates, but it is useful to know how much we can trust them. The variance of the posterior is one way to measure this. The variance of the Beta posterior is given by

$$\text{var}[\theta|\mathcal{D}] = \frac{(a + N_1)(b + N_0)}{(a + N_1 + b + N_0)^2(a + N_1 + b + N_0 + 1)} \tag{3.25}$$

We can simplify this formidable expression in the case that $N \gg a, b$, to get

$$\text{var}[\theta|\mathcal{D}] \approx \frac{N_1 N_0}{NNN} = \frac{\hat{\theta}(1 - \hat{\theta})}{N} \tag{3.26}$$

where $\hat{\theta}$ is the MLE. Hence the "**error bar**" in our estimate (i.e., the posterior standard deviation), is given by

$$\sigma = \sqrt{\text{var}[\theta|\mathcal{D}]} \approx \sqrt{\frac{\hat{\theta}(1 - \hat{\theta})}{N}} \tag{3.27}$$

We see that the uncertainty goes down at a rate of $1/\sqrt{N}$. Note, however, that the uncertainty (variance) is maximized when $\hat{\theta} = 0.5$, and is minimized when $\hat{\theta}$ is close to 0 or 1. This means it is easier to be sure that a coin is biased than to be sure that it is fair.

3.3.4 Posterior predictive distribution

So far, we have been focusing on inference of the unknown parameter(s). Let us now turn our attention to prediction of future observable data.

Consider predicting the probability of heads in a single future trial under a $\mathrm{Beta}(a, b)$ posterior. We have

$$p(\tilde{x} = 1|\mathcal{D}) = \int_0^1 p(x = 1|\theta)p(\theta|\mathcal{D})d\theta \tag{3.28}$$

$$= \int_0^1 \theta \, \mathrm{Beta}(\theta|a, b)d\theta = \mathbb{E}\left[\theta|\mathcal{D}\right] = \frac{a}{a+b} \tag{3.29}$$

Thus we see that the mean of the posterior predictive distribution is equivalent (in this case) to plugging in the posterior mean parameters: $p(\tilde{x}|\mathcal{D}) = \mathrm{Ber}(\tilde{x}|\mathbb{E}\left[\theta|\mathcal{D}\right])$.

3.3.4.1 Overfitting and the black swan paradox

Suppose instead that we plug-in the MLE, i.e., we use $p(\tilde{x}|\mathcal{D}) \approx \mathrm{Ber}(\tilde{x}|\hat{\theta}_{MLE})$. Unfortunately, this approximation can perform quite poorly when the sample size is small. For example, suppose we have seen $N = 3$ tails in a row. The MLE is $\hat{\theta} = 0/3 = 0$, since this makes the observed data as probable as possible. However, using this estimate, we predict that heads are impossible. This is called the **zero count problem** or the **sparse data problem**, and frequently occurs when estimating counts from small amounts of data. One might think that in the era of "big data", such concerns are irrelevant, but note that once we partition the data based on certain criteria — such as the number of times a *specific person* has engaged in a *specific activity* — the sample sizes can become much smaller. This problem arises, for example, when trying to perform personalized recommendation of web pages. Thus Bayesian methods are still useful, even in the big data regime (Jordan 2011).

The zero-count problem is analogous to a problem in philosophy called the **black swan paradox**. This is based on the ancient Western conception that all swans were white. In that context, a black swan was a metaphor for something that could not exist. (Black swans were discovered in Australia by European explorers in the 17th Century.) The term "black swan paradox" was first coined by the famous philosopher of science Karl Popper; the term has also been used as the title of a recent popular book (Taleb 2007). This paradox was used to illustrate the problem of **induction**, which is the problem of how to draw general conclusions about the future from specific observations from the past.

Let us now derive a simple Bayesian solution to the problem. We will use a uniform prior, so $a = b = 1$. In this case, plugging in the posterior mean gives **Laplace's rule of succession**

$$p(\tilde{x} = 1|\mathcal{D}) = \frac{N_1 + 1}{N_1 + N_0 + 2} \tag{3.30}$$

This justifies the common practice of adding 1 to the empirical counts, normalizing and then plugging them in, a technique known as **add-one smoothing**. (Note that plugging in the MAP

parameters would not have this smoothing effect, since the mode has the form $\hat{\theta} = \frac{N_1+a-1}{N+a+b-2}$, which becomes the MLE if $a = b = 1$.)

3.3.4.2 Predicting the outcome of multiple future trials

Suppose now we were interested in predicting the number of heads, x, in M future trials. This is given by

$$p(x|\mathcal{D}, M) = \int_0^1 \text{Bin}(x|\theta, M)\text{Beta}(\theta|a, b)d\theta \tag{3.31}$$

$$= \binom{M}{x} \frac{1}{B(a, b)} \int_0^1 \theta^x (1-\theta)^{M-x} \theta^{a-1}(1-\theta)^{b-1} d\theta \tag{3.32}$$

We recognize the integral as the normalization constant for a $\text{Beta}(a+x, M-x+b)$ distribution. Hence

$$\int_0^1 \theta^x (1-\theta)^{M-x} \theta^{a-1}(1-\theta)^{b-1} d\theta = B(x+a, M-x+b) \tag{3.33}$$

Thus we find that the posterior predictive is given by the following, known as the (compound) **beta-binomial** distribution:

$$Bb(x|a, b, M) \triangleq \binom{M}{x} \frac{B(x+a, M-x+b)}{B(a, b)} \tag{3.34}$$

This distribution has the following mean and variance

$$\mathbb{E}[x] = M\frac{a}{a+b}, \text{ var}[x] = \frac{Mab}{(a+b)^2}\frac{(a+b+M)}{a+b+1} \tag{3.35}$$

If $M = 1$, and hence $x \in \{0, 1\}$, we see that the mean becomes $\mathbb{E}[x|\mathcal{D}] = p(x = 1|\mathcal{D}) = \frac{a}{a+b}$, which is consistent with Equation 3.29.

This process is illustrated in Figure 3.7(a). We start with a Beta(2,2) prior, and plot the posterior predictive density after seeing $N_1 = 3$ heads and $N_0 = 17$ tails. Figure 3.7(b) plots a plug-in approximation using a MAP estimate. We see that the Bayesian prediction has longer tails, spreading its probablity mass more widely, and is therefore less prone to overfitting and blackswan type paradoxes.

3.4 The Dirichlet-multinomial model

In the previous section, we discussed how to infer the probability that a coin comes up heads. In this section, we generalize these results to infer the probability that a dice with K sides comes up as face k. This might seem like another toy exercise, but the methods we will study are widely used to analyse text data, biosequence data, etc., as we will see later.

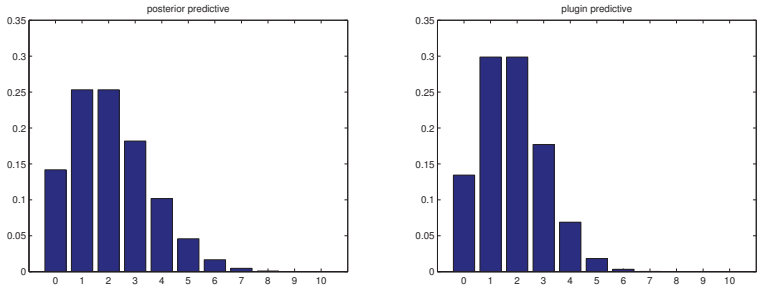


Figure 3.7 (a) Posterior predictive distributions after seeing $N_1 = 3, N_0 = 17$. (b) Plugin approximation. Figure generated by `betaBinomPostPredDemo`.

3.4.1 Likelihood

Suppose we observe N dice rolls, $\mathcal{D} = \{x_1, \ldots, x_N\}$, where $x_i \in \{1, \ldots, K\}$. If we assume the data is iid, the likelihood has the form

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{N_k} \tag{3.36}$$

where $N_k = \sum_{i=1}^{N} \mathbb{I}(y_i = k)$ is the number of times event k occured (these are the sufficient statistics for this model). The likelihood for the multinomial model has the same form, up to an irrelevant constant factor.

3.4.2 Prior

Since the parameter vector lives in the K-dimensional probability simplex, we need a prior that has support over this simplex. Ideally it would also be conjugate. Fortunately, the Dirichlet distribution (Section 2.5.4) satisfies both criteria. So we will use the following prior:

$$\mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \mathbb{I}(\mathbf{x} \in S_K) \tag{3.37}$$

3.4.3 Posterior

Multiplying the likelihood by the prior, we find that the posterior is also Dirichlet:

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{3.38}$$

$$\propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \prod_{k=1}^{K} \theta_k^{\alpha_k + N_k - 1} \tag{3.39}$$

$$= \mathrm{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1, \ldots, \alpha_K + N_K) \tag{3.40}$$

We see that the posterior is obtained by adding the prior hyper-parameters (pseudo-counts) α_k to the empirical counts N_k.

We can derive the mode of this posterior (i.e., the MAP estimate) by using calculus. However, we must enforce the constraint that $\sum_k \theta_k = 1$.[2]. We can do this by using a **Lagrange multiplier**. The constrained objective function, or **Lagrangian**, is given by the log likelihood plus log prior plus the constraint:

$$\ell(\boldsymbol{\theta}, \lambda) = \sum_k N_k \log \theta_k + \sum_k (\alpha_k - 1) \log \theta_k + \lambda \left(1 - \sum_k \theta_k\right) \tag{3.41}$$

To simplify notation, we define $N'_k \triangleq N_k + \alpha_k - 1$. Taking derivatives with respect to λ yields the original constraint:

$$\frac{\partial \ell}{\partial \lambda} = \left(1 - \sum_k \theta_k\right) = 0 \tag{3.42}$$

Taking derivatives with respect to θ_k yields

$$\frac{\partial \ell}{\partial \theta_k} = \frac{N'_k}{\theta_k} - \lambda = 0 \tag{3.43}$$

$$N'_k = \lambda \theta_k \tag{3.44}$$

We can solve for λ using the sum-to-one constraint:

$$\sum_k N'_k = \lambda \sum_k \theta_k \tag{3.45}$$

$$N + \alpha_0 - K = \lambda \tag{3.46}$$

where $\alpha_0 \triangleq \sum_{k=1}^K \alpha_k$ is the equivalent sample size of the prior. Thus the MAP estimate is given by

$$\hat{\theta}_k = \frac{N_k + \alpha_k - 1}{N + \alpha_0 - K} \tag{3.47}$$

which is consistent with Equation 2.77. If we use a uniform prior, $\alpha_k = 1$, we recover the MLE:

$$\hat{\theta}_k = N_k / N \tag{3.48}$$

This is just the empirical fraction of times face k shows up.

2. We do not need to explicitly enforce the constraint that $\theta_k \geq 0$ since the gradient of the objective has the form $N_k/\theta_k - \lambda$; so negative values would reduce the objective, rather than maximize it. (Of course, this does not preclude setting $\theta_k = 0$, and indeed this is the optimal solution if $N_k = 0$ and $\alpha_k = 1$.)

3.4.4 Posterior predictive

The posterior predictive distribution for a single multinoulli trial is given by the following expression:

$$p(X=j|\mathcal{D}) = \int p(X=j|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{3.49}$$

$$= \int p(X=j|\theta_j)\left[\int p(\boldsymbol{\theta}_{-j},\theta_j|\mathcal{D})d\boldsymbol{\theta}_{-j}\right]d\theta_j \tag{3.50}$$

$$= \int \theta_j p(\theta_j|\mathcal{D})d\theta_j = \mathbb{E}\left[\theta_j|\mathcal{D}\right] = \frac{\alpha_j+N_j}{\sum_k(\alpha_k+N_k)} = \frac{\alpha_j+N_j}{\alpha_0+N} \tag{3.51}$$

where $\boldsymbol{\theta}_{-j}$ are all the components of $\boldsymbol{\theta}$ except θ_j. See also Exercise 3.13.

The above expression avoids the zero-count problem, just as we saw in Section 3.3.4.1. In fact, this form of Bayesian smoothing is even more important in the multinomial case than the binary case, since the likelihood of data sparsity increases once we start partitioning the data into many categories.

3.4.4.1 Worked example: language models using bag of words

One application of Bayesian smoothing using the Dirichlet-multinomial model is to **language modeling**, which means predicting which words might occur next in a sequence. Here we will take a very simple-minded approach, and assume that the i'th word, $X_i \in \{1,\ldots,K\}$, is sampled independently from all the other words using a $\mathrm{Cat}(\boldsymbol{\theta})$ distribution. This is called the **bag of words** model. Given a past sequence of words, how can we predict which one is likely to come next?

For example, suppose we observe the following sequence (part of a children's nursery rhyme):

```
Mary had a little lamb, little lamb, little lamb,
Mary had a little lamb, its fleece as white as snow
```

Furthermore, suppose our vocabulary consists of the following words:

```
mary lamb little big fleece white black snow rain unk
1    2    3      4   5      6     7     8    9    10
```

Here **unk** stands for unknown, and represents all other words that do not appear elsewhere on the list. To encode each line of the nursery rhyme, we first strip off punctuation, and remove any **stop words** such as "a", "as", "the", etc. We can also perform **stemming**, which means reducing words to their base form, such as stripping off the final *s* in plural words, or the *ing* from verbs (e.g., *running* becomes *run*). In this example, no words need stemming. Finally, we replace each word by its index into the vocabulary to get:

```
1 10 3 2 3 2 3 2
1 10 3 2 10 5 10 6 8
```

We now ignore the word order, and count how often each word occurred, resulting in a histogram of word counts:

Token	1	2	3	4	5	6	7	8	9	10
Word	mary	lamb	little	big	fleece	white	black	snow	rain	unk
Count	2	4	4	0	1	1	0	1	0	4

Denote the above counts by N_j. If we use a $\mathrm{Dir}(\boldsymbol{\alpha})$ prior for $\boldsymbol{\theta}$, the posterior predictive is just

$$p(\tilde{X} = j|D) = E[\theta_j|D] = \frac{\alpha_j + N_j}{\sum_{j'} \alpha_{j'} + N_{j'}} = \frac{1 + N_j}{10 + 17} \tag{3.52}$$

If we set $\alpha_j = 1$, we get

$$p(\tilde{X} = j|D) = (3/27, 5/27, 5/27, 1/27, 2/27, 2/27, 1/27, 2/27, 1/27, 5/27) \tag{3.53}$$

The modes of the predictive distribution are $X = 2$ ("lamb") and $X = 10$ ("unk"). Note that the words "big", "black" and "rain" are predicted to occur with non-zero probability in the future, even though they have never been seen before. Later on we will see more sophisticated language models.

3.5 Naive Bayes classifiers

In this section, we discuss how to classify vectors of discrete-valued features, $\mathbf{x} \in \{1, \ldots, K\}^D$, where K is the number of values for each feature, and D is the number of features. We will use a generative approach. This requires us to specify the class conditional distribution, $p(\mathbf{x}|y = c)$. The simplest approach is to assume the features are **conditionally independent** given the class label. This allows us to write the class conditional density as a product of one dimensional densities:

$$p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} p(x_j|y = c, \boldsymbol{\theta}_{jc}) \tag{3.54}$$

The resulting model is called a **naive Bayes classifier** (NBC).

The model is called "naive" since we do not expect the features to be independent, even conditional on the class label. However, even if the naive Bayes assumption is not true, it often results in classifiers that work well (Domingos and Pazzani 1997). One reason for this is that the model is quite simple (it only has $O(CD)$ parameters, for C classes and D features), and hence it is relatively immune to overfitting.

The form of the class-conditional density depends on the type of each feature. We give some possibilities below:

- In the case of real-valued features, we can use the Gaussian distribution: $p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{N}(x_j|\mu_{jc}, \sigma_{jc}^2)$, where μ_{jc} is the mean of feature j in objects of class c, and σ_{jc}^2 is its variance.
- In the case of binary features, $x_j \in \{0, 1\}$, we can use the Bernoulli distribution: $p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} \mathrm{Ber}(x_j|\mu_{jc})$, where μ_{jc} is the probability that feature j occurs in class c. This is sometimes called the **multivariate Bernoulli naive Bayes** model. We will see an application of this below.

- In the case of categorical features, $x_j \in \{1, \ldots, K\}$, we can model use the multinoulli distribution: $p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} \mathrm{Cat}(x_j|\boldsymbol{\mu}_{jc})$, where $\boldsymbol{\mu}_{jc}$ is a histogram over the K possible values for x_j in class c.

Obviously we can handle other kinds of features, or use different distributional assumptions. Also, it is easy to mix and match features of different types.

3.5.1 Model fitting

We now discuss how to "train" a naive Bayes classifier. This usually means computing the MLE or the MAP estimate for the parameters. However, we will also discuss how to compute the full posterior, $p(\boldsymbol{\theta}|\mathcal{D})$.

3.5.1.1 MLE for NBC

The probability for a single data case is given by

$$p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = p(y_i|\boldsymbol{\pi}) \prod_j p(x_{ij}|\boldsymbol{\theta}_j) = \prod_c \pi_c^{\mathbb{I}(y_i=c)} \prod_j \prod_c p(x_{ij}|\boldsymbol{\theta}_{jc})^{\mathbb{I}(y_i=c)} \tag{3.55}$$

Hence the log-likelihood is given by

$$\log p(\mathcal{D}|\boldsymbol{\theta}) \quad = \quad \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij}|\boldsymbol{\theta}_{jc}) \tag{3.56}$$

We see that this expression decomposes into a series of terms, one concerning $\boldsymbol{\pi}$, and DC terms containing the $\boldsymbol{\theta}_{jc}$'s. Hence we can optimize all these parameters separately.

From Equation 3.48, the MLE for the class prior is given by

$$\hat{\pi}_c \quad = \quad \frac{N_c}{N} \tag{3.57}$$

where $N_c \triangleq \sum_i \mathbb{I}(y_i = c)$ is the number of examples in class c.

The MLE for the likelihood depends on the type of distribution we choose to use for each feature. For simplicity, let us suppose all features are binary, so $x_j|y = c \sim \mathrm{Ber}(\theta_{jc})$. In this case, the MLE becomes

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c} \tag{3.58}$$

It is extremely simple to implement this model fitting procedure: See Algorithm 8 for some pseudo-code (and `naiveBayesFit` for some Matlab code). This algorithm obviously takes $O(ND)$ time. The method is easily generalized to handle features of mixed type. This simplicity is one reason the method is so widely used.

Figure 3.8 gives an example where we have 2 classes and 600 binary features, representing the presence or absence of words in a bag-of-words model. The plot visualizes the $\boldsymbol{\theta}_c$ vectors for the two classes. The big spike at index 107 corresponds to the word "subject", which occurs in both classes with probability 1. (In Section 3.5.4, we discuss how to "filter out" such uninformative features.)

Algorithm 3.1: Fitting a naive Bayes classifier to binary features

```
N_c = 0, N_jc = 0;
for i = 1 : N do
    c = y_i // Class label of i'th example;
    N_c := N_c + 1 ;
    for j = 1 : D do
        if x_ij = 1 then
            N_jc := N_jc + 1
π̂_c = N_c / N, θ̂_jc = N_jc / N
```

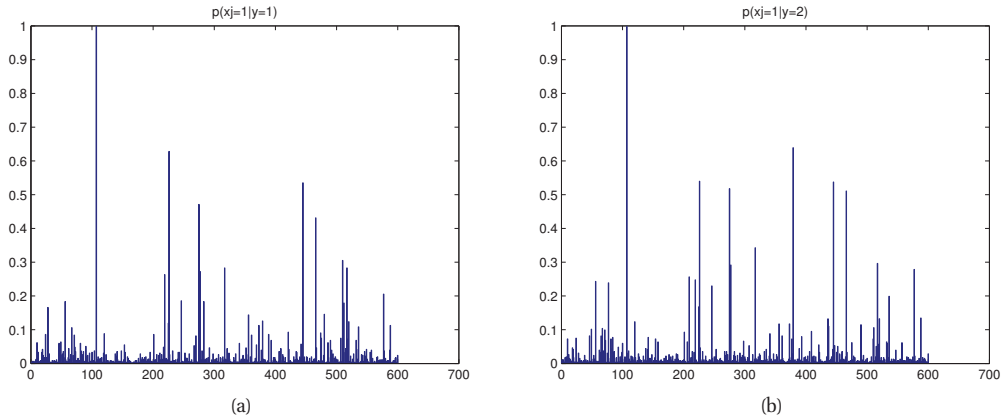


Figure 3.8 Class conditional densities $p(x_j = 1|y = c)$ for two document classes, corresponding to "X windows" and "MS windows". Figure generated by `naiveBayesBowDemo`.

3.5.1.2 Bayesian naive Bayes

The trouble with maximum likelihood is that it can overfit. For example, consider the example in Figure 3.8: the feature corresponding to the word "subject" (call it feature j) always occurs in both classes, so we estimate $\hat{\theta}_{jc} = 1$. What will happen if we encounter a new email which does not have this word in it? Our algorithm will crash and burn, since we will find that $p(y = c|\mathbf{x}, \hat{\boldsymbol{\theta}}) = 0$ for both classes! This is another manifestation of the black swan paradox discussed in Section 3.3.4.1.

A simple solution to overfitting is to be Bayesian. For simplicity, we will use a factored prior:

$$p(\boldsymbol{\theta}) \;=\; p(\boldsymbol{\pi}) \prod_{j=1}^{D} \prod_{c=1}^{C} p(\theta_{jc}) \tag{3.59}$$

We will use a $\mathrm{Dir}(\boldsymbol{\alpha})$ prior for $\boldsymbol{\pi}$ and a $\mathrm{Beta}(\beta_0, \beta_1)$ prior for each θ_{jc}. Often we just take $\boldsymbol{\alpha} = \mathbf{1}$ and $\boldsymbol{\beta} = \mathbf{1}$, corresponding to add-one or Laplace smoothing.

Combining the factored likelihood in Equation 3.56 with the factored prior above gives the following factored posterior:

$$p(\boldsymbol{\theta}|\mathcal{D}) = p(\boldsymbol{\pi}|\mathcal{D}) \prod_{j=1}^{D} \prod_{c=}^{C} p(\theta_{jc}|\mathcal{D}) \tag{3.60}$$

$$p(\boldsymbol{\pi}|\mathcal{D}) = \mathrm{Dir}(N_1 + \alpha_1 \ldots, N_C + \alpha_C) \tag{3.61}$$

$$p(\theta_{jc}|\mathcal{D}) = \mathrm{Beta}((N_c - N_{jc}) + \beta_0, N_{jc} + \beta_1) \tag{3.62}$$

In other words, to compute the posterior, we just update the prior counts with the empirical counts from the likelihood. It is straightforward to modify algorithm 8 to handle this version of model "fitting".

3.5.2 Using the model for prediction

At test time, the goal is to compute

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto p(y = c|\mathcal{D}) \prod_{j=1}^{D} p(x_j|y = c, \mathcal{D}) \tag{3.63}$$

The correct Bayesian procedure is to integrate out the unknown parameters:

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto \left[\int \mathrm{Cat}(y = c|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\mathcal{D}) d\boldsymbol{\pi}\right] \tag{3.64}$$

$$\prod_{j=1}^{D} \left[\int \mathrm{Ber}(x_j|y = c, \theta_{jc}) p(\boldsymbol{\theta}_{jc}|\mathcal{D})\right] \tag{3.65}$$

Fortunately, this is easy to do, at least if the posterior is Dirichlet. In particular, from Equation 3.51, we know the posterior predictive density can be obtained by simply plugging in the posterior mean parameters $\overline{\boldsymbol{\theta}}$. Hence

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto \overline{\pi}_c \prod_{j=1}^{D} (\overline{\theta}_{jc})^{\mathbb{I}(x_j=1)} (1 - \overline{\theta}_{jc})^{\mathbb{I}(x_j=0)} \tag{3.66}$$

$$\overline{\theta}_{jk} = \frac{N_{jc} + \beta_1}{N_c + \beta_0 + \beta_1} \tag{3.67}$$

$$\overline{\pi}_c = \frac{N_c + \alpha_c}{N + \alpha_0} \tag{3.68}$$

where $\alpha_0 = \sum_c \alpha_c$.

If we have approximated the posterior by a single point, $p(\boldsymbol{\theta}|\mathcal{D}) \approx \delta_{\hat{\boldsymbol{\theta}}}(\boldsymbol{\theta})$, where $\hat{\boldsymbol{\theta}}$ may be the ML or MAP estimate, then the posterior predictive density is obtained by simply plugging in the parameters, to yield a virtually identical rule:

$$p(y = c|\mathbf{x}, \mathcal{D}) \propto \hat{\pi}_c \prod_{j=1}^{D} (\hat{\theta}_{jc})^{\mathbb{I}(x_j=1)} (1 - \hat{\theta}_{jc})^{\mathbb{I}(x_j=0)} \tag{3.69}$$

The only difference is we replaced the posterior mean $\overline{\boldsymbol{\theta}}$ with the posterior mode or MLE $\hat{\boldsymbol{\theta}}$. However, this small difference can be important in practice, since the posterior mean will result in less overfitting (see Section 3.4.4.1).

3.5.3 The log-sum-exp trick

We now discuss one important practical detail that arises when using generative classifiers of any kind. We can compute the posterior over class labels using Equation 2.13, using the appropriate class-conditional density (and a plug-in approximation). Unfortunately a naive implementation of Equation 2.13 can fail due to **numerical underflow**. The problem is that $p(\mathbf{x}|y=c)$ is often a very small number, especially if $\mathbf{x}$ is a high-dimensional vector. This is because we require that $\sum_{\mathbf{x}} p(\mathbf{x}|y) = 1$, so the probability of observing any particular high-dimensional vector is small. The obvious solution is to take logs when applying Bayes rule, as follows:

$$\log p(y=c|\mathbf{x}) = b_c - \log\left[\sum_{c'=1}^{C} e^{b_{c'}}\right] \tag{3.70}$$

$$b_c \triangleq \log p(\mathbf{x}|y=c) + \log p(y=c) \tag{3.71}$$

However, this requires evaluating the following expression

$$\log[\sum_{c'} e^{b_{c'}}] = \log \sum_{c'} p(y=c', \mathbf{x}) = \log p(\mathbf{x}) \tag{3.72}$$

and we can't add up in the log domain. Fortunately, we can factor out the largest term, and just represent the remaining numbers relative to that. For example,

$$\log(e^{-120} + e^{-121}) = \log\left(e^{-120}(e^0 + e^{-1})\right) = \log(e^0 + e^{-1}) - 120 \tag{3.73}$$

In general, we have

$$\log \sum_c e^{b_c} = \log\left[(\sum_c e^{b_c - B})e^B\right] = \left[\log(\sum_c e^{b_c - B})\right] + B \tag{3.74}$$

where $B = \max_c b_c$. This is called the **log-sum-exp** trick, and is widely used. (See the function `logsumexp` for an implementation.)

This trick is used in Algorithm 1 which gives pseudo-code for using an NBC to compute $p(y_i|\mathbf{x}_i, \hat{\boldsymbol{\theta}})$. See `naiveBayesPredict` for the Matlab code. Note that we do not need the log-sum-exp trick if we only want to compute $\hat{y}_i$, since we can just maximize the unnormalized quantity $\log p(y_i = c) + \log p(\mathbf{x}_i|y=c)$.

3.5.4 Feature selection using mutual information

Since an NBC is fitting a joint distribution over potentially many features, it can suffer from overfitting. In addition, the run-time cost is $O(D)$, which may be too high for some applications.

One common approach to tackling both of these problems is to perform **feature selection**, to remove "irrelevant" features that do not help much with the classification problem. The simplest approach to feature selection is to evaluate the relevance of each feature separately, and then

Algorithm 3.2: Predicting with a naive bayes classifier for binary features

```
for i = 1 : N do
    for c = 1 : C do
        L_ic = log π̂_c;
        for j = 1 : D do
            if x_ij = 1 then L_ic := L_ic + log θ̂_jc else L_ic := L_ic + log(1 − θ̂_jc)
    p_ic = exp(L_ic − logsumexp(L_i,:));
    ŷ_i = argmax_c p_ic;
```

take the top K, where K is chosen based on some tradeoff between accuracy and complexity. This approach is known as variable **ranking**, **filtering**, or **screening**.

One way to measure relevance is to use mutual information (Section 2.8.3) between feature X_j and the class label Y:

$$I(X, Y) = \sum_{x_j} \sum_{y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)p(y)} \tag{3.75}$$

The mutual information can be thought of as the reduction in entropy on the label distribution once we observe the value of feature j. If the features are binary, it is easy to show (Exercise 3.21) that the MI can be computed as follows

$$I_j \quad = \quad \sum_{c} \left[\theta_{jc} \pi_c \log \frac{\theta_{jc}}{\theta_j} + (1 - \theta_{jc}) \pi_c \log \frac{1 - \theta_{jc}}{1 - \theta_j} \right] \tag{3.76}$$

where $\pi_c = p(y = c)$, $\theta_{jc} = p(x_j = 1|y = c)$, and $\theta_j = p(x_j = 1) = \sum_c \pi_c \theta_{jc}$. (All of these quantities can be computed as a by-product of fitting a naive Bayes classifier.)

Figure 3.1 illustrates what happens if we apply this to the binary bag of words dataset used in Figure 3.8. We see that the words with highest mutual information are much more discriminative than the words which are most probable. For example, the most probable word in both classes is "subject", which always occurs because this is newsgroup data, which always has a subject line. But obviously this is not very discriminative. The words with highest MI with the class label are (in decreasing order) "windows", "microsoft", "DOS" and "motif", which makes sense, since the classes correspond to Microsoft Windows and X Windows.

3.5.5 Classifying documents using bag of words

Document classification is the problem of classifying text documents into different categories. One simple approach is to represent each document as a binary vector, which records whether each word is present or not, so $x_{ij} = 1$ iff word j occurs in document i, otherwise $x_{ij} = 0$. We can then use the following class conditional density:

$$p(\mathbf{x}_i|y_i = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} \text{Ber}(x_{ij}|\theta_{jc}) = \prod_{j=1}^{D} \theta_{jc}^{\mathbb{I}(x_{ij})} (1 - \theta_{jc})^{\mathbb{I}(1-x_{ij})} \tag{3.77}$$

class 1	prob	class 2	prob	highest MI	MI
subject	0.998	subject	0.998	windows	0.215
this	0.628	windows	0.639	microsoft	0.095
with	0.535	this	0.540	dos	0.092
but	0.471	with	0.538	motif	0.078
you	0.431	but	0.518	window	0.067

Table 3.1 We list the 5 most likely words for class 1 (X windows) and class 2 (MS windows). We also show the 5 words with highest mutual information with class label. Produced by `naiveBayesBowDemo`

This is called the **Bernoulli product model**, or the **binary independence model**.

However, ignoring the number of times each word occurs in a document loses some information (McCallum and Nigam 1998). A more accurate representation counts the number of occurrences of each word. Specifically, let $\mathbf{x}_i$ be a vector of counts for document i, so $x_{ij} \in \{0, 1, \ldots, N_i\}$, where N_i is the number of terms in document i (so $\sum_{j=1}^{D} x_{ij} = N_i$). For the class conditional densities, we can use a multinomial distribution:

$$p(\mathbf{x}_i|y_i = c, \boldsymbol{\theta}) = \text{Mu}(\mathbf{x}_i|N_i, \boldsymbol{\theta}_c) = \frac{N_i!}{\prod_{j=1}^{D} x_{ij}!} \prod_{j=1}^{D} \theta_{jc}^{x_{ij}} \tag{3.78}$$

where we have implicitly assumed that the document length N_i is independent of the class. Here θ_{jc} is the probability of generating word j in documents of class c; these parameters satisfy the constraint that $\sum_{j=1}^{D} \theta_{jc} = 1$ for each class c.[3]

Although the multinomial classifier is easy to train and easy to use at test time, it does not work particularly well for document classification. One reason for this is that it does not take into account the **burstiness** of word usage. This refers to the phenomenon that most words never appear in any given document, but if they do appear once, they are likely to appear more than once, i.e., words occur in bursts.

The multinomial model cannot capture the burstiness phenomenon. To see why, note that Equation 3.78 has the form $\theta_{jc}^{N_{ij}}$, and since $\theta_{jc} \ll 1$ for rare words, it becomes increasingly unlikely to generate many of them. For more frequent words, the decay rate is not as fast. To see why intuitively, note that the most frequent words are function words which are not specific to the class, such as "and", "the", and "but"; the chance of the word "and" occuring is pretty much the same no matter how many time it has previously occurred (modulo document length), so the independence assumption is more reasonable for common words. However, since rare words are the ones that matter most for classification purposes, these are the ones we want to model the most carefully.

Various ad hoc heuristics have been proposed to improve the performance of the multinomial document classifier (Rennie et al. 2003). We now present an alternative class conditional density that performs as well as these ad hoc methods, yet is probabilistically sound (Madsen et al. 2005).

3. Since Equation 3.78 models each word independently, this model is often called a naive Bayes classifier, although technically the features x_{ij} are not independent, because of the constraint $\sum_j x_{ij} = N_i$.

Suppose we simply replace the multinomial class conditional density with the **Dirichlet Compound Multinomial** or **DCM** density, defined as follows:

$$p(\mathbf{x}_i|y_i = c, \boldsymbol{\alpha}) = \int \mathrm{Mu}(\mathbf{x}_i|N_i, \boldsymbol{\theta}_c)\mathrm{Dir}(\boldsymbol{\theta}_c|\boldsymbol{\alpha}_c)d\boldsymbol{\theta}_c = \frac{N_i!}{\prod_{j=1}^{D} x_{ij}!}\frac{B(\mathbf{x}_i + \boldsymbol{\alpha}_c)}{B(\boldsymbol{\alpha}_c)} \tag{3.79}$$

(This equation is derived in Equation 5.24.) Surprisingly this simple change is all that is needed to capture the burstiness phenomenon. The intuitive reason for this is as follows: After seeing one occurence of a word, say word j, the posterior counts on θ_j gets updated, making another occurence of word j more likely. By contrast, if θ_j is fixed, then the occurences of each word are independent. The multinomial model corresponds to drawing a ball from an urn with K colors of ball, recording its color, and then replacing it. By contrast, the DCM model corresponds to drawing a ball, recording its color, and then replacing it with one additional copy; this is called the **Polya urn**.

Using the DCM as the class conditional density gives much better results than using the multinomial, and has performance comparable to state of the art methods, as described in (Madsen et al. 2005). The only disadvantage is that fitting the DCM model is more complex; see (Minka 2000e; Elkan 2006) for the details.

Exercises

Exercise 3.1 MLE for the Bernoulli/ binomial model

Derive Equation 3.22 by optimizing the log of the likelihood in Equation 3.11.

Exercise 3.2 Marginal likelihood for the Beta-Bernoulli model

In Equation 5.23, we showed that the marginal likelihood is the ratio of the normalizing constants:

$$p(D) = \frac{Z(\alpha_1 + N_1, \alpha_0 + N_0)}{Z(\alpha_1, \alpha_0)} = \frac{\Gamma(\alpha_1 + N_1)\Gamma(\alpha_0 + N_0)}{\Gamma(\alpha_1 + \alpha_0 + N)}\frac{\Gamma(\alpha_1 + \alpha_0)}{\Gamma(\alpha_1)\Gamma(\alpha_0)} \tag{3.80}$$

We will now derive an alternative derivation of this fact. By the chain rule of probability,

$$p(x_{1:N}) = p(x_1)p(x_2|x_1)p(x_3|x_{1:2})\ldots \tag{3.81}$$

In Section 3.3.4, we showed that the posterior predictive distribution is

$$p(X = k|D_{1:N}) = \frac{N_k + \alpha_k}{\sum_i N_i + \alpha_i} \triangleq \frac{N_k + \alpha_k}{N + \alpha} \tag{3.82}$$

where $k \in \{0, 1\}$ and $D_{1:N}$ is the data seen so far. Now suppose $D = H, T, T, H, H$ or $D = 1, 0, 0, 1, 1$. Then

$$p(D) = \frac{\alpha_1}{\alpha}\cdot\frac{\alpha_0}{\alpha+1}\cdot\frac{\alpha_0+1}{\alpha+2}\cdot\frac{\alpha_1+1}{\alpha+3}\cdot\frac{\alpha_1+2}{\alpha+4} \tag{3.83}$$

$$= \frac{[\alpha_1(\alpha_1+1)(\alpha_1+2)]\,[\alpha_0(\alpha_0+1)]}{\alpha(\alpha+1)\cdots(\alpha+4)} \tag{3.84}$$

$$= \frac{[(\alpha_1)\cdots(\alpha_1+N_1-1)]\,[(\alpha_0)\cdots(\alpha_0+N_0-1)]}{(\alpha)\cdots(\alpha+N-1)} \tag{3.85}$$

Show how this reduces to Equation 3.80 by using the fact that, for integers, $(\alpha - 1)! = \Gamma(\alpha)$.

Exercise 3.3 Posterior predictive for Beta-Binomial model

Recall from Equation 3.32 that the posterior predictive for the Beta-Binomial is given by

$$p(x|n, D) = Bb(x|\alpha'_0, \alpha'_1, n) \tag{3.86}$$

$$= \frac{B(x+\alpha'_1, n-x+\alpha'_0)}{B(\alpha'_1, \alpha'_0)} \binom{n}{x} \tag{3.87}$$

Prove that this reduces to

$$p(\tilde{x} = 1|D) = \frac{\alpha'_1}{\alpha'_0 + \alpha'_1} \tag{3.88}$$

when $n = 1$ (and hence $x \in \{0, 1\}$). i.e., show that

$$Bb(1|\alpha'_1, \alpha'_0, 1) = \frac{\alpha'_1}{\alpha'_1 + \alpha'_0} \tag{3.89}$$

Hint: use the fact that

$$\Gamma(\alpha_0 + \alpha_1 + 1) = (\alpha_0 + \alpha_1 + 1)\Gamma(\alpha_0 + \alpha_1) \tag{3.90}$$

Exercise 3.4 Beta updating from censored likelihood

(Source: Gelman.) Suppose we toss a coin $n = 5$ times. Let X be the number of heads. We observe that there are fewer than 3 heads, but we don't know exactly how many. Let the prior probability of heads be $p(\theta) = \text{Beta}(\theta|1, 1)$. Compute the posterior $p(\theta|X < 3)$ up to normalization constants, i.e., derive an expression proportional to $p(\theta, X < 3)$. Hint: the answer is a mixture distribution.

Exercise 3.5 Uninformative prior for log-odds ratio

Let

$$\phi = \text{logit}(\theta) = \log \frac{\theta}{1-\theta} \tag{3.91}$$

Show that if $p(\phi) \propto 1$, then $p(\theta) \propto \text{Beta}(\theta|0, 0)$. Hint: use the change of variables formula.

Exercise 3.6 MLE for the Poisson distribution

The Poisson pmf is defined as $\text{Poi}(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$, for $x \in \{0, 1, 2, \ldots\}$ where $\lambda > 0$ is the rate parameter. Derive the MLE.

Exercise 3.7 Bayesian analysis of the Poisson distribution

In Exercise 3.6, we defined the Poisson distribution with rate λ and derived its MLE. Here we perform a conjugate Bayesian analysis.

a. Derive the posterior $p(\lambda|D)$ assuming a conjugate prior $p(\lambda) = \text{Ga}(\lambda|a, b) \propto \lambda^{a-1}e^{-\lambda b}$. Hint: the posterior is also a Gamma distribution.

b. What does the posterior mean tend to as $a \to 0$ and $b \to 0$? (Recall that the mean of a $\text{Ga}(a, b)$ distribution is a/b.)

Exercise 3.8 MLE for the uniform distribution

(Source: Kaelbling.) Consider a uniform distribution centered on 0 with width $2a$. The density function is given by

$$p(x) = \frac{1}{2a} I(x \in [-a, a]) \tag{3.92}$$

a. Given a data set $x_1, \ldots, x_n$, what is the maximum likelihood estimate of a (call it $\hat{a}$)?

b. What probability would the model assign to a new data point x_{n+1} using $\hat{a}$?

c. Do you see any problem with the above approach? Briefly suggest (in words) a better approach.

Exercise 3.9 Bayesian analysis of the uniform distribution

Consider the uniform distribution $\text{Unif}(0, \theta)$. The maximum likelihood estimate is $\hat{\theta} = \max(\mathcal{D})$, as we saw in Exercise 3.8, but this is unsuitable for predicting future data since it puts zero probability mass outside the training data. In this exercise, we will perform a Bayesian analysis of the uniform distribution (following (Minka 2001a)). The conjugate prior is the Pareto distribution, $p(\theta) = \text{Pareto}(\theta|b, K)$, defined in Section 2.4.6. Given a Pareto prior, the joint distribution of θ and $\mathcal{D} = (x_1, \ldots, x_N)$ is

$$p(\mathcal{D}, \theta) = \frac{Kb^K}{\theta^{N+K+1}} \mathbb{I}(\theta \geq \max(\mathcal{D})) \tag{3.93}$$

Let $m = \max(\mathcal{D})$. The evidence (the probability that all N samples came from the same uniform distribution) is

$$p(\mathcal{D}) = \int_m^\infty \frac{Kb^K}{\theta^{N+K+1}} d\theta \tag{3.94}$$

$$= \begin{cases} \frac{K}{(N+K)b^N} & \text{if } m \leq b \\ \frac{Kb^K}{(N+K)m^{N+K}} & \text{if } m > b \end{cases} \tag{3.95}$$

Derive the posterior $p(\theta|\mathcal{D})$, and show that if can be expressed as a Pareto distribution.

Exercise 3.10 Taxicab (tramcar) problem

Suppose you arrive in a new city and see a taxi numbered 100. How many taxis are there in this city? Let us assume taxis are numbered sequentially as integers starting from 0, up to some unknown upper bound θ. (We number taxis from 0 for simplicity; we can also count from 1 without changing the analysis.) Hence the likelihood function is $p(x) = U(0, \theta)$, the uniform distribution. The goal is to estimate θ. We will use the Bayesian analysis from Exercise 3.9.

a. Suppose we see one taxi numbered 100, so $\mathcal{D} = \{100\}$, $m = 100$, $N = 1$. Using an (improper) non-informative prior on θ of the form $p(\theta) = Pa(\theta|0, 0) \propto 1/\theta$, what is the posterior $p(\theta|D)$?

b. Compute the posterior mean, mode and median number of taxis in the city, if such quantities exist.

c. Rather than trying to compute a point estimate of the number of taxis, we can compute the predictive density over the next taxicab number using

$$p(D'|D, \alpha) = \int p(D'|\theta) p(\theta|D, \alpha) d\theta = p(D'|\beta) \tag{3.96}$$

where $\alpha = (b, K)$ are the hyper-parameters, $\beta = (c, N + K)$ are the updated hyper-parameters. Now consider the case $D = \{m\}$, and $D' = \{x\}$. Using Equation 3.95, write down an expression for

$$p(x|D, \alpha) \tag{3.97}$$

As above, use a non-informative prior $b = K = 0$.

d. Use the predictive density formula to compute the probability that the next taxi you will see (say, the next day) has number 100, 50 or 150, i.e., compute $p(x = 100|D, \alpha)$, $p(x = 50|D, \alpha)$, $p(x = 150|D, \alpha)$.

e. Briefly describe (1-2 sentences) some ways we might make the model more accurate at prediction.

Exercise 3.11 Bayesian analysis of the exponential distribution

A lifetime X of a machine is modeled by an exponential distribution with unknown parameter θ. The likelihood is $p(x|\theta) = \theta e^{-\theta x}$ for $x \geq 0$, $\theta > 0$.

a. Show that the MLE is $\hat{\theta} = 1/\overline{x}$, where $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$.

b. Suppose we observe $X_1 = 5, X_2 = 6, X_3 = 4$ (the lifetimes (in years) of 3 different iid machines). What is the MLE given this data?

c. Assume that an expert believes θ should have a prior distribution that is also exponential

$$p(\theta) = \text{Expon}(\theta|\lambda) \tag{3.98}$$

Choose the prior parameter, call it $\hat{\lambda}$, such that $\mathbb{E}[\theta] = 1/3$. Hint: recall that the Gamma distribution has the form

$$\text{Ga}(\theta|a, b) \quad \propto \quad \theta^{a-1} e^{-\theta b} \tag{3.99}$$

and its mean is a/b.

d. What is the posterior, $p(\theta|\mathcal{D}, \hat{\lambda})$?

e. Is the exponential prior conjugate to the exponential likelihood?

f. What is the posterior mean, $\mathbb{E}\left[\theta|\mathcal{D}, \hat{\lambda}\right]$?

g. Explain why the MLE and posterior mean differ. Which is more reasonable in this example?

Exercise 3.12 MAP estimation for the Bernoulli with non-conjugate priors

(Source: Jaakkola.) In the book, we discussed Bayesian inference of a Bernoulli rate parameter with the prior $p(\theta) = \text{Beta}(\theta|\alpha, \beta)$. We know that, with this prior, the MAP estimate is given by

$$\hat{\theta} = \frac{N_1 + \alpha - 1}{N + \alpha + \beta - 2} \tag{3.100}$$

where N_1 is the number of heads, N_0 is the number of tails, and $N = N_0 + N_1$ is the total number of trials.

a. Now consider the following prior, that believes the coin is fair, or is slightly biased towards tails:

$$p(\theta) \quad = \quad \begin{cases} 0.5 & \text{if } \theta = 0.5 \\ 0.5 & \text{if } \theta = 0.4 \\ 0 & \text{otherwise} \end{cases} \tag{3.101}$$

Derive the MAP estimate under this prior as a function of N_1 and N.

b. Suppose the true parameter is $\theta = 0.41$. Which prior leads to a better estimate when N is small? Which prior leads to a better estimate when N is large?

Exercise 3.13 Posterior predictive distribution for a batch of data with the dirichlet-multinomial model

In Equation 3.51, we gave the the posterior predictive distribution for a single multinomial trial using a dirichlet prior. Now consider predicting a *batch* of new data, $\tilde{\mathcal{D}} = (X_1, \ldots, X_m)$, consisting of m single multinomial trials (think of predicting the next m words in a sentence, assuming they are drawn iid). Derive an expression for

$$p(\tilde{\mathcal{D}}|\mathcal{D}, \alpha) \tag{3.102}$$

Your answer should be a function of $\boldsymbol{\alpha}$, and the old and new counts (sufficient statistics), defined as

$$N_k^{old} = \sum_{i \in \mathcal{D}} I(x_i = k) \tag{3.103}$$

$$N_k^{new} = \sum_{i \in \tilde{\mathcal{D}}} I(x_i = k) \tag{3.104}$$

Hint: recall that, for a vector of counts, $N_{1:K}$, the marginal likelihood (evidence) is given by

$$p(\mathcal{D}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \prod_k \frac{\Gamma(N_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{3.105}$$

where $\alpha = \sum_k \alpha_k$ and $N = \sum_k N_k$.

Exercise 3.14 Posterior predictive for Dirichlet-multinomial

(Source: Koller.).

a. Suppose we compute the empirical distribution over letters of the Roman alphabet plus the space character (a distribution over 27 values) from 2000 samples. Suppose we see the letter "e" 260 times. What is $p(x_{2001} = e|\mathcal{D})$, if we assume $\boldsymbol{\theta} \sim \text{Dir}(\alpha_1, \ldots, \alpha_{27})$, where $\alpha_k = 10$ for all k?

b. Suppose, in the 2000 samples, we saw "e" 260 times, "a" 100 times, and "p" 87 times. What is $p(x_{2001} = p, x_{2002} = a|\mathcal{D})$, if we assume $\boldsymbol{\theta} \sim \text{Dir}(\alpha_1, \ldots, \alpha_{27})$, where $\alpha_k = 10$ for all k? Show your work.

Exercise 3.15 Setting the beta hyper-parameters

Suppose $\theta \sim \beta(\alpha_1, \alpha_2)$ and we believe that $\mathbb{E}[\theta] = m$ and $\text{var}[\theta] = v$. Using Equation 2.62, solve for α_1 and α_2 in terms of m and v. What values do you get if $m = 0.7$ and $v = 0.2^2$?

Exercise 3.16 Setting the beta hyper-parameters II

(Source: Draper.) Suppose $\theta \sim \beta(\alpha_1, \alpha_2)$ and we believe that $\mathbb{E}[\theta] = m$ and $p(\ell < \theta < u) = 0.95$. Write a program that can solve for α_1 and α_2 in terms of m, ℓ and u. Hint: write α_2 as a function of α_1 and m, so the pdf only has one unknown; then write down the probability mass contained in the interval as an integral, and minimize its squared discrepancy from 0.95. What values do you get if $m = 0.15$, $\ell = 0.05$ and $u = 0.3$? What is the equivalent sample size of this prior?

Exercise 3.17 Marginal likelihood for beta-binomial under uniform prior

Suppose we toss a coin N times and observe N_1 heads. Let $N_1 \sim \text{Bin}(N, \theta)$ and $\theta \sim \text{Beta}(1, 1)$. Show that the marginal likelihood is $p(N_1|N) = 1/(N+1)$. Hint: $\Gamma(x+1) = x!$ if x is an integer.

Exercise 3.18 Bayes factor for coin tossing

Suppose we toss a coin $N = 10$ times and observe $N_1 = 9$ heads. Let the null hypothesis be that the coin is fair, and the alternative be that the coin can have any bias, so $p(\theta) = \text{Unif}(0, 1)$. Derive the Bayes factor $BF_{1,0}$ in favor of the biased coin hypothesis. What if $N = 100$ and $N_1 = 90$? Hint: see Exercise 3.17.

Exercise 3.19 Irrelevant features with naive Bayes

(Source: Jaakkola.) Let $x_{iw} = 1$ if word w occurs in document i and $x_{iw} = 0$ otherwise. Let θ_{cw} be the estimated probability that word w occurs in documents of class c. Then the log-likelihood that document

$\mathbf{x}$ belongs to class c is

$$\log p(\mathbf{x}_i|c,\theta) = \log \prod_{w=1}^{W} \theta_{cw}^{x_{iw}} (1-\theta_{cw})^{1-x_{iw}} \tag{3.106}$$

$$= \sum_{w=1}^{W} x_{iw} \log \theta_{cw} + (1-x_{iw}) \log(1-\theta_{cw}) \tag{3.107}$$

$$= \sum_{w=1}^{W} x_{iw} \log \frac{\theta_{cw}}{1-\theta_{cw}} + \sum_{w} \log(1-\theta_{cw}) \tag{3.108}$$

where W is the number of words in the vocabulary. We can write this more succintly as

$$\log p(\mathbf{x}_i|c,\theta) = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\beta}_c \tag{3.109}$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{iW})$ is a bit vector, $\boldsymbol{\phi}(\mathbf{x}_i) = (\mathbf{x}_i, 1)$, and

$$\boldsymbol{\beta}_c = (\log \frac{\theta_{c1}}{1-\theta_{c1}}, \ldots, \log \frac{\theta_{cW}}{1-\theta_{cW}}, \sum_{w} \log(1-\theta_{cw}))^T \tag{3.110}$$

We see that this is a linear classifier, since the class-conditional density is a linear function (an inner product) of the parameters $\boldsymbol{\beta}_c$.

a. Assuming $p(C=1) = p(C=2) = 0.5$, write down an expression for the log posterior odds ratio, $\log_2 \frac{p(c=1|\mathbf{x}_i)}{p(c=2|\mathbf{x}_i)}$, in terms of the features $\boldsymbol{\phi}(\mathbf{x}_i)$ and the parameters $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$.

b. Intuitively, words that occur in both classes are not very "discriminative", and therefore should not affect our beliefs about the class label. Consider a particular word w. State the conditions on $\theta_{1,w}$ and $\theta_{2,w}$ (or equivalently the conditions on $\beta_{1,w}, \beta_{2,w}$) under which the presence or absence of w in a test document will have no effect on the class posterior (such a word will be ignored by the classifier). Hint: using your previous result, figure out when the posterior odds ratio is 0.5/0.5.

c. The posterior mean estimate of θ, using a Beta(1,1) prior, is given by

$$\hat{\theta}_{cw} = \frac{1 + \sum_{i \in c} x_{iw}}{2 + n_c} \tag{3.111}$$

where the sum is over the n_c documents in class c. Consider a particular word w, and suppose it always occurs in every document (regardless of class). Let there be n_1 documents of class 1 and n_2 be the number of documents in class 2, where $n_1 \neq n_2$ (since e.g., we get much more non-spam than spam; this is an example of class imbalance). If we use the above estimate for θ_{cw}, will word w be ignored by our classifier? Explain why or why not.

d. What other ways can you think of which encourage "irrelevant" words to be ignored?

Exercise 3.20 Class conditional densities for binary data

Consider a generative classifier for C classes with class conditional density $p(\mathbf{x}|y)$ and uniform class prior $p(y)$. Suppose all the D features are binary, $x_j \in \{0,1\}$. If we assume all the features are conditionally independent (the naive Bayes assumption), we can write

$$p(\mathbf{x}|y=c) = \prod_{j=1}^{D} \mathrm{Ber}(x_j|\theta_{jc}) \tag{3.112}$$

This requires DC parameters.

a. Now consider a different model, which we will call the "full" model, in which all the features are fully dependent (i.e., we make no factorization assumptions). How might we represent $p(\mathbf{x}|y=c)$ in this case? How many parameters are needed to represent $p(\mathbf{x}|y=c)$?

b. Assume the number of features D is fixed. Let there be N training cases. If the sample size N is very small, which model (naive Bayes or full) is likely to give lower test set error, and why?

c. If the sample size N is very large, which model (naive Bayes or full) is likely to give lower test set error, and why?

d. What is the computational complexity of fitting the full and naive Bayes models as a function of N and D? Use big-Oh notation. (Fitting the model here means computing the MLE or MAP parameter estimates. You may assume you can convert a D-bit vector to an array index in $O(D)$ time.)

e. What is the computational complexity of applying the full and naive Bayes models at test time to a single test case?

f. Suppose the test case has missing data. Let $\mathbf{x}_v$ be the visible features of size v, and $\mathbf{x}_h$ be the hidden (missing) features of size h, where $v+h=D$. What is the computational complexity of computing $p(y|\mathbf{x}_v, \hat{\boldsymbol{\theta}})$ for the full and naive Bayes models, as a function of v and h?

Exercise 3.21 Mutual information for naive Bayes classifiers with binary features

Derive Equation 3.76.

Exercise 3.22 Fitting a naive bayes spam filter by hand

(Source: Daphne Koller.). Consider a Naive Bayes model (multivariate Bernoulli version) for spam classification with the vocabulary V="secret", "offer", "low", "price", "valued", "customer", "today", "dollar", "million", "sports", "is", "for", "play", "healthy", "pizza". We have the following example spam messages "million dollar offer", "secret offer today", "secret is secret" and normal messages, "low price for valued customer", "play secret sports today", "sports is healthy", "low price pizza". Give the MLEs for the following parameters: θ_{spam}, $\theta_{\text{secret}|\text{spam}}$, $\theta_{\text{secret}|\text{non-spam}}$, $\theta_{\text{sports}|\text{non-spam}}$, $\theta_{\text{dollar}|\text{spam}}$.