CHAPTER 8

NONLINEAR BELIEF MODELS

While linear models will always remain an important tool for approximating functions (remember we mean that they are linear in the parameters), there are some problems where we cannot avoid models that are nonlinear in the parameters. Some examples include:

EXAMPLE 8.1

We wish to estimate the probability that a customer will purchase an online product with characterized by price p (which is a decision variable) and other features of the product which we denote by $a_1, \ldots, a_K$. We can then describe our product using

$$x = (p, a_1, \ldots, a_K).$$

We think of x as a decision, although the only controllable parameter is p. We may feel that we can model the probability of a sale using a logistic model of the form

$$p(x) = \frac{e^{U(x|\theta)}}{1 + e^{U(x|\theta)}},$$

where $U(x|\theta)$ is a linear function of independent variables with the general form

$$U(x|\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots.$$

We start with an example of a relatively simple nonlinear problem that arises in the context of designing bidding policies. We then transition to more general strategies that can be applied to a wide range of problems.

8.1 AN OPTIMAL BIDDING PROBLEM

Imagine that industrial customers come to you requesting price quotes on contracts to provide a product of some type. This might be laptops for a large company, a component for a product that the customer is building (such as disk drives for the laptops), or materials used in a manufacturing process. You have to quote a price, recognizing that you have to match or beat competing prices. If you win the contract, you might be left wondering if you could have asked for a little more. If you lose the contact, you are kicking yourself if you feel that a slightly lower price would have helped you win the contract.

Figure 8.1 illustrates the economics of different prices. There is a breakeven price p^b, below which you will lose money on the contract. Then there is a price point $\bar{p}_c$ for each customer c. If you quote a price above this number, you will lose the contract. The problem, of course, is that you do not know $\bar{p}_c$. As the figure illustrates, increasing the price above p^b produces dramatic increases in profits, especially for competitive, commodity products.

We are going to assume that we have several opportunities to quote a price and observe a buy/not-buy decision from the customer. We are not allowed to observe $\bar{p}_c$ directly, even after the fact. After each iteration of quoting a price and observing a decision of whether or not to accept the bid, we have an opportunity to use what we have learned before deciding on our next bid. We know that we have to bid prices above p^b, and we know what prices have been accepted in the past. If we discover that we can get higher prices, the impact on our profits can be significant. Our challenge is trying to learn $\bar{p}_c$ while balancing the profits we realize against the information gained while trying higher prices (and potentially losing some contracts).

We can also consider a version of this problem from the customer's point of view. Suppose that there are now multiple sellers offering a single product. A customer makes a bid for the product at a price of his or her choosing. The bid is successful if there is a seller willing to accept it, but the customer does not get to observe the sellers' willingness to sell before making the bid. In this case, higher bids are more likely to be successful, but the customer can save money by finding the optimal price. Bidding too low and failing to secure an offer carries an opportunity cost (perhaps we are forced to wait a period of time before bidding again). We might call this the "Priceline problem," because of its similarities to the business model of the well-known online travel agency.

Overall, we can see a clear potential for optimal learning in the bidding problem. In fact, variations of this problem have attracted a great deal of attention in the revenue management community, precisely from an optimal learning perspective. However, it is less clear precisely how learning should be applied. The problem has a number

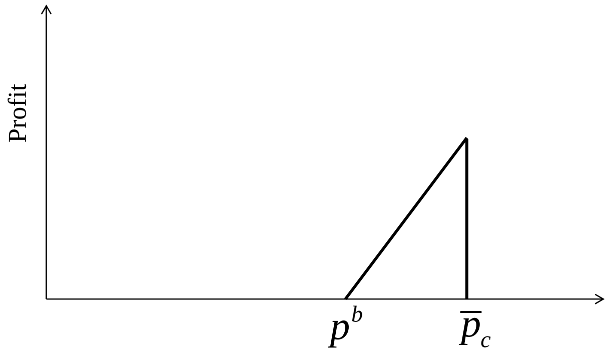


Figure 8.1 Illustration of the value of quoting a price p that is greater than the break even price p^b but lower than the cuttoff price $\bar{p}_c$ for customer c.

of features that bring us outside the scope of the clean, fundamental models presented in Chapters 3 and 5. For example, in the bidding problem, we are not able to observe an unbiased sample of the truth. That is, in the industrial setting, the company does not get to observe the exact amount that the client was willing to pay for the contract, only whether or not the quoted price was accepted. As we will see in this chapter, this creates an additional challenge for our analysis. We are no longer able to make use of an elegant conjugate prior, as in Chapter 2.

New challenges call for new methods. In contrast with our earlier focus on knowledge gradient methods, we look at a simpler greedy policy for the bidding problem. However, this policy is still in line with our general approach to optimal learning. We will use a Bayesian model to represent our uncertainty about customer demand. Our policy incorporates this uncertainty into the decision-making. As a result, we will tend to quote higher prices to the customers than we would without the learning dimension. How much higher will depend on the amount of uncertainty we have. By doing so, we will take on more risk with the first few contracts, but the information we collect will help us to make money in the long run. The bidding problem shows that optimal learning provides value in a slightly messier setting that goes beyond the standard models we have discussed up to this point.

8.1.1 Modeling customer demand

Suppose that we are working with a sequence of bids. The customers' behavior is modeled using the concept of *valuation* or *willingness to pay*. The nth customer values a product or service at W^n. We make the sale as long as this valuation is at least as much as our quoted price p, that is, $W^n \geq p$. Our revenue, in this case, is the price p. If $W^n < p$, our revenue is zero. Thus, our *expected revenue*, for a fixed price p, is given by

$$R(p) = p \cdot P(W^n \geq p). \tag{8.1}$$

In most applications, we will work with the *expected profit*

$$P(p) = (p - c) \cdot P(W^n \geq p), \tag{8.2}$$

where c is the cost of the product to the seller.

In real life, many customers may not have an exact number for the most they are willing to pay. In any case, we will never observe this quantity, only whether it is larger or smaller than p. However, the idea of the valuation gives us a way to think about the bidding problem formally and put a number on the probability that a price will be accepted. Initially, let us assume that the valuations $W^0, W^1, ...$ are independent and identically distributed. Thus, while different customers can have different valuations, all customers come from the same population. This may be a reasonable assumption if a type of industrial contract serves a particular market (such as semiconductor manufacturers). Later, in Section 8.1.3, we begin to push the boundaries of this assumption.

8.1.2 Some valuation models

Nearly any standard distribution might be used to model customer valuation. We list several distributions that have been used in the literature on bidding. We will make use of some of these models to illustrate some of the issues inherent in the problem, although our main focus will be the logistic model of Section 8.1.3. Many of these same distributions also appeared in Chapter 2 as sampling models.

Uniform valuation The simplest model assumes that the valuation follows a uniform distribution, $W^n \sim U[a,b]$. It follows that the probability of making the sale is

$$P(W^n \geq p) = \begin{cases} 1 & p < a \\ \frac{b}{b-a} - \frac{p}{b-a} & p \in [a,b] \\ 0 & p > b \end{cases},$$

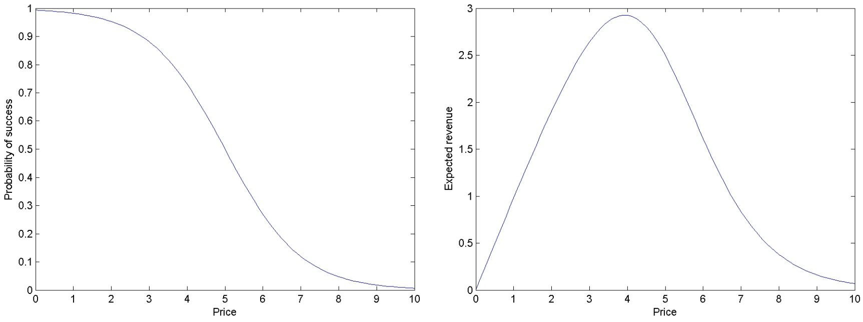
which is a linear function of p. This is known as *linear demand*. If we assume that $a = 0$, the only parameter in this model is the maximum valuation b. If this parameter is unknown, a natural choice for a distribution of belief would be a Pareto prior (see Section 2.6.3).

The uniform valuation is clean, but involves several strong assumptions. We are assuming that there is a fair amount of variation; customers are equally likely to have low or high valuations. Additionally, we assume that there is a cutoff point. A high enough price is guaranteed to lose the sale.

Exponential valuation An exponential valuation assumes that $W^n \sim Exp(\lambda)$. The demand curve then has the simple form $P(W^n \geq p) = e^{-\lambda p}$. As in the linear model, higher prices are less likely to be successful. However, now any price will have a non-zero probability of being successful. We are assuming that there will be a small proportion of customers willing to pay very large prices.

Lognormal valuation In the lognormal model, we assume that

$$P(W^n \geq p) = 1 - \Phi\left(\frac{\log p - \mu}{\sigma}\right),$$



(a) (b)

Figure 8.2 Example of (a) probability of making a sale and (b) expected revenue under a logistic model with $\mu_1 = 5$, $\mu_2 = 1$.

where μ and σ are the parameters of the demand curve. This model implies that $\log W^n \sim \mathcal{N}\left(\mu, \sigma^2\right)$. If we had a way to observe the exact values W^n, we could put a normal prior on μ and treat $\log W^n$ as a normal observation, enabling the use of the normal-normal learning model.

8.1.3 The logit model

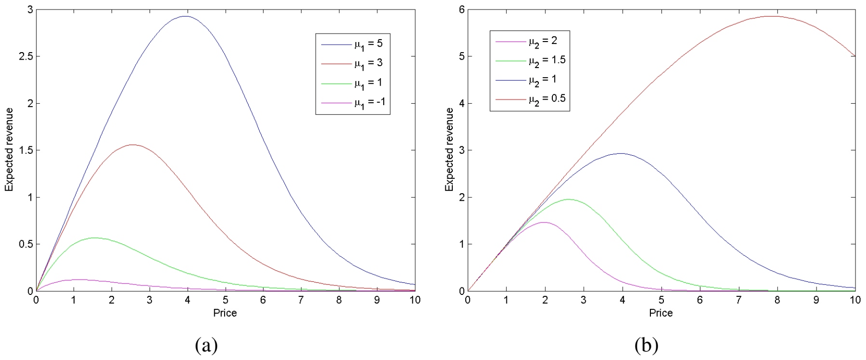
The logit model is a particularly attractive method of approximating the probability that a customer will accept a bid. The logic model expresses the probability of making a sale as

$$P\left(W^n \geq p\right) = \frac{1}{1 + e^{-(\mu_1 - \mu_2 p)}}. \tag{8.3}$$

When $\mu_2 > 0$, plotting (8.3) as a function of p yields a logistic curve, a well-known mathematical model for predicting demand, population growth, technology adoption, and other cyclical phenomena. Figure 8.2(a) gives an example for a particular choice of μ_1 and μ_2. We see a classic S-curve, flipped around so that higher prices lead to lower success probabilities. Balancing the decreasing success probability with the increasing potential revenue, the expected revenue function (8.1) has a maximum at approximately $p^* \approx 3.9$.

The parameters μ_1 and μ_2 determine the customer's reaction to different price offers. We can view μ_1 as the *market share* of the seller. Even if the seller were to give away the product for free, the probability of success would only be $\frac{1}{1+e^{-\mu_1}}$, that is, some customers would still prefer to buy from a competitor. Essentially, these customers have a negative valuation of our product, and there is no way we could convince them to buy it. The higher the value of μ_1, the higher the market share and the closer $P\left(W^n \geq 0\right)$ is to 1.

The second parameter μ_2 can be viewed as the *price sensitivity* of the customers. Large values of μ_2 tend to make the curve in Figure 8.3 steeper, causing the probability of success to decrease faster as the price goes up. We typically require that $\mu_2 > 0$, to preserve the S-curve shape of the success probability. Negative values of μ_2 would



(a) (b)

Figure 8.3 Sensitivity of the revenue curve to changes in (a) market share and (b) price parameter.

imply that higher prices are more likely to be successful, which does not make sense for our problem.

One important advantage of the logistic model is that it allows us to move beyond the assumption that the customers come from the same population. We could, for example, allow the demand curve to depend on attributes of the customer, as given by

$$P\left(W^n \geq p\right) = \frac{1}{1 + e^{-\left(\mu^T x^n - p\mu_{P+1}\right)}}. \tag{8.4}$$

Here, $x^n = \left[x_1^n, x_2^n, ..., x_P^n\right]^T$ is a vector representing P attributes of the customer. For example, the attributes could reflect the location of the customer and its size. The vector μ contains the parameters assigned to the attributes. We also have a single price sensitivity parameter $\mu_{P+1} > 0$. This model is an example of the *logistic regression* technique in statistics, which fits a set of parameters to a success probability. Throughout this chapter, we mostly focus on the simple two-parameter model of (8.3), but it is important to remember that our analysis can easily be extended to the multi-parameter case.

Optimal learning comes into play when we, as the seller, do not know the exact values of the parameters μ_1, μ_2. We may have some prior estimates of these parameters, based on past sales figures. However, what makes this problem especially difficult is that even small changes to the parameters will greatly change the expected revenue function. Figure 8.3(a) shows the expected revenue curve for different values of μ_1, with μ_2 fixed at 1. Figure 8.3(b) shows the same curve for different values of μ_2, with μ_1 fixed at 5.

Higher values of μ_1 move the optimal price to the right, but they also expand the magnitude of the entire revenue curve. Increasing μ_1 from 1 to 3 moves the optimal price roughly from 1.5 to 2.5, but triples the expected revenue collected in the process. Smaller values of μ_2 (that is, closer to zero) have the same two effects, but move the optimal price more than they increase optimal revenue. When we allow both parameters to change at the same time, as in Figure 8.4, even small changes will allow for a wide range of possible optimal prices and revenues.

8.1.4 Bayesian modeling for dynamic pricing

We will use a Bayesian model to represent our uncertainty about the parameters. In this setting, it becomes especially important to construct our prior in such a way as to realistically cover our range of uncertainty about the optimal price. Furthermore, the nature of the observations in this problem creates additional challenges. We are not able to observe the customer valuations W^n. Rather, we only observe whether or not $W^n \geq p$. The clean, conjugate Bayesian models introduced in Chapter 2 cannot be directly applied, and we need to do additional work to be able to use them.

8.1.4.1 A conjugate prior for choosing between two demand curves

Before we delve into the intricacies of non-conjugate Bayesian learning, let us first begin with a simple, stylized model that allows for a conjugate prior. In dynamic pricing, our observations are binary: either the customer accepts the offer (that is, $W^n \geq p$), or not. One way to obtain a conjugate prior is if the truth is binary, as well. Suppose that there are only two possible demand curves. Each curve has known, fixed parameters, but we do not know which curve is the right one for describing customer valuations. Figure 8.5 gives an example where we are trying to choose between a uniform valuation on the interval $[3, 7]$, and an exponential valuation with parameter 0.2.

This model is somewhat stylized, but may be useful in some cases. It may be that we, as the seller, have a large amount of historical data on sales figures, enough to fit any particular type of demand curve. We could conduct a statistical analysis to fit a uniform valuation model, or an exponential model, or perhaps a logistic model. For each model, we would fit a different set of parameters. However, we are not sure which type of demand curve is most appropriate. A logistic model may be fundamentally better-suited to the data than a uniform model. In that case, the binary-truth problem may help us to distinguish between two competing types of demand models.

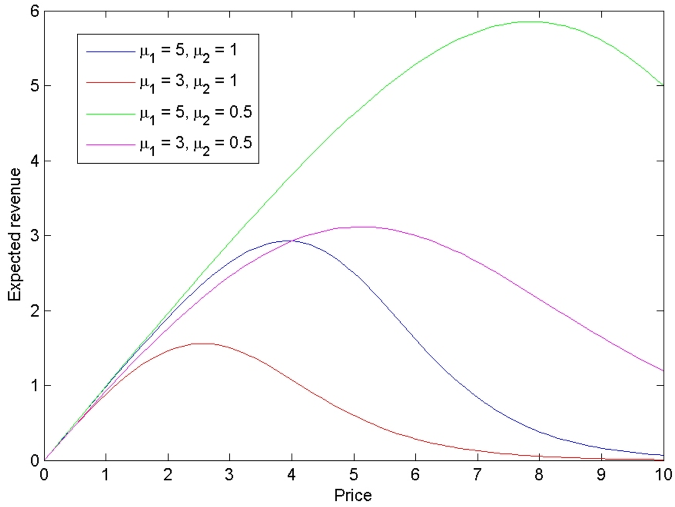


Figure 8.4 Sensitivity of the revenue curve to changes in both parameters.

Let f_1 and f_2 be the two demand curves under consideration. That is, $f_i(p) = P(W^n \geq p)$ under two different models $i = 1, 2$. Let A be the event that f_1 is the correct demand curve; then, A^c is the event that f_2 is correct. We begin with a prior probability $q^0 = P^0(A)$. For a fixed price p, the revenue function is given by

$$R(p) = p\left[q^0 f_1(p) + \left(1 - q^0\right) f_2(p)\right]. \tag{8.5}$$

We then make a pricing decision p^0 and observe either $W^1 \geq p^0$ or $W^1 < p^0$. First, let us consider the case where $W^1 \geq p^0$, that is, we make the sale with price p^0. Using Bayes' rule, we can derive

$$P\left(A \,|\, W^1 \geq p^0\right) = \frac{P\left(W^1 \geq p^0 \,|\, A\right) P(A)}{P\left(W^1 \geq p^0 \,|\, A\right) P(A) + P\left(W^1 \geq p^0 \,|\, A^c\right) P(A^c)}. \tag{8.6}$$

Observe that

$$\begin{aligned} P\left(W^1 \geq p^0 \,|\, A\right) &= f_1(p), \\ P\left(W^1 \geq p^0 \,|\, A^c\right) &= f_2(p), \\ P(A) &= q^0. \end{aligned}$$

We can let $q^1 = P\left(A \,|\, W^1 \geq p^0\right)$ denote our posterior probability of the event A. Then, (8.6) becomes

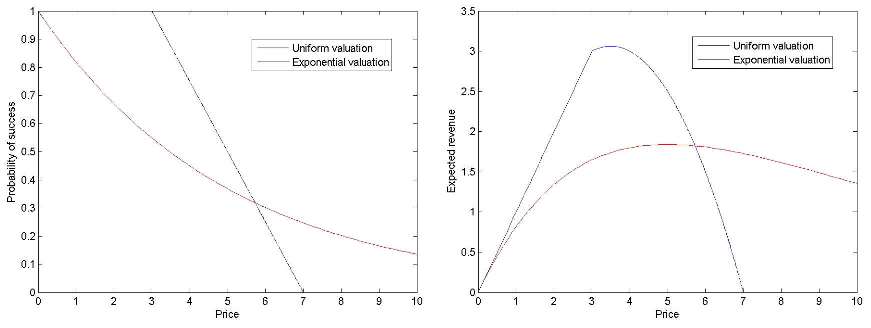
$$q^1 = \frac{q^0 f_1(p)}{q^0 f_1(p) + (1 - q^0) f_2(p)}.$$

Repeating the same analysis for the event that $W^1 < p^0$ produces the updating equation

$$q^1 = \frac{q^0 (1 - f_1(p))}{q^0 (1 - f_1(p)) + (1 - q^0)(1 - f_2(p))}.$$

Let $X^n = I_{\{W^n \geq p^n\}}$. That is, $X^n = 1$ if our price p^n is successful, and $X^n = 0$ otherwise. Then, we obtain a clean updating formula

$$q^{n+1} = \frac{q^n f_1(p^n)^{X^{n+1}} (1 - f_1(p^n))^{1 - X^{n+1}}}{q^n f_1(p^n)^{X^{n+1}} (1 - f_1(p^n))^{1 - X^{n+1}} + (1 - q^n) f_2(p^n)^{X^{n+1}} (1 - f_2(p^n))^{1 - X^{n+1}}}. \tag{8.7}$$



(a) (b)

Figure 8.5 Examples of (a) demand curves and (b) revenue curves for a pricing problem with two truths.

This may be the simplest possible conjugate model. We put a simple discrete (actually binary) distribution of belief on the probability that the truth is given by a particular demand curve. When we make a binary observation, the posterior distribution is also discrete.

We might then apply a variety of techniques to make our pricing decision. A common approach in the literature is to use a simple myopic policy,

$$p^n = \arg\max_p \mathbb{E}R(p) = \arg\max_p p\left[q^0 f_1(p) + \left(1-q^0\right) f_2(p)\right].$$

We could also apply some of the ideas from Section 4.7.1, where the observation was also binary, to this problem. There is one important issue to keep in mind, however. Observe that Figure 8.5(a) has a point where the two demand curves intersect. That is, there is a price $\bar{p}$ for which $f_1(\bar{p}) = f_2(\bar{p}) = \bar{f}$. Substituting this price into (8.7) gives us

$$\begin{aligned} q^{n+1} &= \frac{q^n \bar{f}^{X^{n+1}} \left(1-\bar{f}\right)^{1-X^{n+1}}}{q^n \bar{f}^{X^{n+1}} \left(1-\bar{f}\right)^{1-X^{n+1}} + (1-q^n)\bar{f}^{X^{n+1}} \left(1-\bar{f}\right)^{1-X^{n+1}}} \\ &= \frac{q^n \bar{f}^{X^{n+1}} \left(1-\bar{f}\right)^{1-X^{n+1}}}{\bar{f}^{X^{n+1}} \left(1-\bar{f}\right)^{1-X^{n+1}}} \\ &= q^n. \end{aligned}$$

If our policy chooses the price $\bar{p}$, our beliefs about $P(A)$ will remain unchanged. Furthermore, since the policy determines what to do based on our beliefs, it will continue to choose the same price $\bar{p}$ thereafter. We would thus get stuck with the same beliefs forever. For that reason, the price $\bar{p}$ is known as the *non-informative price* or the *confounding price*.

Fortunately, since we have both demand curves specified exactly (we just don't know which is the right one), we can calculate the confounding price before we set out to solve the problem. The literature suggests a simple fix for the problem of confounding. We simply fix another price $\tilde{p} \neq \bar{p}$ beforehand. If our policy tells us to choose the price $\bar{p}$, we choose $\tilde{p}$ instead. Otherwise, we follow the policy.

8.1.4.2 Moment matching for non-conjugate problems*

Although the simple model of Section 8.1.4.1 has its uses, we are really interested in the case where we do not know the parameters of the demand curve. In any case, before we settle on two demand curves to choose from, we first need to find good parameters for those curves. So, we might focus on one particular type of curve, and then try to learn the right parameters for that type. This will lead us to the problem of non-conjugacy.

Let us illustrate this issue using a simple example. Suppose that the customers' valuations are drawn from a uniform distribution on the interval $[0, B]$, where B is unknown. Recalling Chapter 2, a natural choice of prior distribution on B is the Pareto distribution. If $B \sim Pareto\left(\alpha^0, b^0\right)$, and if we were able to observe the exact valuation $W^1 \sim U[0, B]$, we could apply the conjugate update

$$\begin{aligned} \alpha^1 &= \alpha^0 + 1 \\ b^1 &= \max\left(b^0, W^1\right). \end{aligned}$$

Unfortunately, we are never able to see the exact valuation. However, we can still derive a posterior distribution of belief, given the incomplete information that we do observe. For simplicity, let us assume that we set a price $p^0 < b^0$. We will first consider the case where $W^1 < p^0$, that is, we lost the sale. We apply Bayes' rule and write

$$g\left(u \,|\, W^1 < p^0\right) = \frac{P\left(W^1 < p^0 \,|\, B = u\right) g\left(u\right)}{P\left(W^1 < p^0\right)}. \tag{8.8}$$

Conditionally given $B = u$, the valuation W^1 has a uniform distribution on $[0, u]$. Thus,

$$P\left(W^1 < p^0 \,|\, B = u\right) = \begin{cases} \frac{p^0}{u} & p^0 < u \\ 1 & p^0 \geq u. \end{cases}$$

Next, the likelihood that $B = u$ is given by the Pareto density,

$$g\left(u\right) = \frac{\alpha^0 \left(b^0\right)^{\alpha^0}}{u^{\alpha^0+1}} \qquad \text{for } u > b^0.$$

Thus, in our calculations, we are implicitly assuming that $u > b^0$, since $B > b^0$ by the definition of a Pareto distribution. Since we are also assuming $p^0 < b^0$, it follows that $p^0 < u$ for all possible u. The numerator of the right-hand side of (8.8) thus becomes

$$P\left(W^1 < p^0 \,|\, B = u\right) g\left(u\right) = \frac{\alpha^0 p^0 \left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}}. \tag{8.9}$$

Integrating this expression from b^0 to infinity, we obtain the denominator of the right-hand side of (8.8),

$$P\left(W^1 < p^0\right) = \int_{b^0}^{\infty} \frac{\alpha^0 p^0 \left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}} du = \frac{\alpha^0}{\alpha^0+1} \frac{p^0}{b^0}. \tag{8.10}$$

Dividing (8.9) by (8.10) yields

$$g\left(u \,|\, W^1 < p^0\right) = \frac{\left(\alpha^0 + 1\right) \left(b^0\right)^{\alpha^0+1}}{u^{\alpha^0+2}}.$$

This is a Pareto density with parameters $\alpha^1 = \alpha^0 + 1$ and $b^1 = b^0$. That is, if we lose the sale, conjugacy is maintained. The conjugate updating equations hint at this. From (8.8), we see that $b^1 = \max\left(b^0, W^1\right)$, if we could observe the exact valuation W^1. However, we are assuming that $p^0 < b^0$. Thus, if we observe $W^1 < p^0$, it follows automatically that $\max\left(b^0, W^1\right) = b^0$. We do not really need to know the exact value of W^1 in this case. For any value of W^1 less than b^0, we will not change the scale parameter of our distribution of belief.

In the other case, however, we run into trouble. Suppose that $W^1 \geq p^0$, with $p^0 < b^0$ as before. Then,

$$P\left(W^1 \geq p^0 \,|\, B = u\right) = \begin{cases} 1 - \frac{p^0}{u} & p^0 < u \\ 0 & p^0 \geq u \end{cases}$$

and

$$P\left(W^1 \geq p^0 \,|\, B = u\right) g\left(u\right) = \frac{\alpha^0 \left(b^0\right)^{\alpha^0}}{u^{\alpha^0+1}} - \frac{\alpha^0 p^0 \left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}} \qquad u > b^0.$$

Integrating this quantity from b^0 to infinity yields

$$P\left(W^1 \geq p^0\right) = 1 - \frac{\alpha^0}{\alpha^0 + 1}\frac{p^0}{b^0},$$

and the posterior density turns out to be

$$g\left(u \,|\, W^1 < p^0\right) = \frac{\frac{\alpha^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+1}} - \frac{\alpha^0 p^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}}}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}} \qquad u > b^0. \tag{8.11}$$

What are we to do with this strange expression? It does not correspond to any standard density. (The difference of two Pareto densities is *not* the density of the difference of two Pareto random variables!) It is certainly not a Pareto density.

We get around this problem by using a technique called *moment-matching*. Define a random variable U that has the density given by (8.11). As long as we have the density, we can compute the first two moments of U as follows. First,

$$\begin{aligned} \mathbb{E}U &= \int_{b^0}^{\infty} u \frac{\frac{\alpha^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+1}} - \frac{\alpha^0 p^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}}}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}} \\ &= \frac{\frac{\alpha^0 b^0}{\alpha^0-1} - p^0}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}}. \end{aligned} \tag{8.12}$$

Similarly,

$$\begin{aligned} \mathbb{E}\left(U^2\right) &= \int_{b^0}^{\infty} u^2 \frac{\frac{\alpha^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+1}} - \frac{\alpha^0 p^0\left(b^0\right)^{\alpha^0}}{u^{\alpha^0+2}}}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}} \\ &= \frac{\frac{\alpha^0\left(b^0\right)^2}{\alpha^0-2} - \frac{\alpha^0 b^0 p^0}{\alpha^0-1}}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}}. \end{aligned} \tag{8.13}$$

Let Y be a random variable following a Pareto distribution with parameters α^1 and b^1. The first and second moments of Y are given by

$$\mathbb{E}Y = \frac{\alpha^1 b^1}{\alpha^1 - 1}, \qquad \mathbb{E}\left(Y^2\right) = \frac{\alpha^1\left(b^1\right)^2}{\alpha^1 - 2}.$$

Moment matching works by setting the moments of Y equal to the moments of U, and solving for α^1 and b^1 in terms of α^0 and b^0. Essentially, we are forcing the posterior distribution to be Pareto, and choosing α^1 and b^1 to make this Pareto distribution resemble the actual posterior distribution, at least up to the first two moments. To do this, we have to solve two sets of nonlinear equations

$$\begin{aligned} \frac{\alpha^1 b^1}{\alpha^1 - 1} &= \frac{\frac{\alpha^0 b^0}{\alpha^0-1} - p^0}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}}, \\ \frac{\alpha^1\left(b^1\right)^2}{\alpha^1 - 2} &= \frac{\frac{\alpha^0\left(b^0\right)^2}{\alpha^0-2} - \frac{\alpha^0 b^0 p^0}{\alpha^0-1}}{1 - \frac{\alpha^0}{\alpha^0+1}\frac{p^0}{b^0}}. \end{aligned}$$

The solution gives us the non-conjugate updating equations,

$$\begin{aligned}\alpha^1 &= 1+\sqrt{\frac{\mathbb{E}\left(U^2\right)}{\mathbb{E}\left(U^2\right)-\left(\mathbb{E}U\right)^2}},\\ b^1 &= \frac{\alpha^1-2}{\alpha^1-1}\frac{\mathbb{E}\left(U^2\right)}{\mathbb{E}U},\end{aligned}$$

where $\mathbb{E}U$ and $\mathbb{E}\left(U^2\right)$ are given in (8.12) and (8.13) in terms of α^0 and b^0. Notice that, for compactness, the updating equation for b^1 is written in terms of α^1. To use these equations, we should first compute α^1, then use that value to find b^1.

Keep in mind that we have not computed updating equations for the case where $p^0 \geq b^0$. In this case, neither success nor failure will give us a Pareto posterior, and we have to repeat the above analysis. Choosing a different demand curve and prior distribution would also require a new round of moment-matching, and most likely new sets of nonlinear equations to solve. Moment-matching is no easy task. However, it is the simplest way to obtain clean updating equations in a problem where the observations are incomplete, that is, we can only obtain imperfect information about the customer valuations.

8.1.5 An approximation for the logit model

The issue of non-conjugacy is also present in the logit model. There is no natural choice of prior for the logistic regression parameters μ_1, μ_2 in (8.3). That is, there is no clean prior distribution that is conjugate with observations sampled from the logistic distribution. The simplest choice of prior, from the decision-maker's point of view, is a multivariate normal distribution on (μ_1, μ_2). Especially if we extend our model to incorporate customer attributes, as in (8.4), a multivariate normal prior would allow us to include correlations in our beliefs about the parameters of different attributes. We followed this same approach in Chapter 7 when we fit a linear regression model to our observations. As we saw earlier, the linear regression model admits an elegant conjugate normal-normal learning model.

Unfortunately, there is no direct analog in the case of logistic regression, because our observations are now binary, of the form

$$X^n = \begin{cases} 1 & \text{If } W^n \geq p^n, \\ 0 & \text{Otherwise.} \end{cases}$$

Recall that $X^n = 1$ if we make the sale, and $X^n = 0$ if we do not. If we start with a normal prior on the logistic parameters, then see an observation of this form, the posterior distribution will not be normal. However, there is an approximation that gives us a set of recursive updating equations. Like we did in (8.1.4.2) with moment matching, we can use this approximation to force the posterior to be normal. If $\mu = (\mu_1, \mu_2)$ has a multivariate normal distribution with mean vector $\bar{\theta}^n = \left(\bar{\theta}^n_1, \bar{\theta}^n_2\right)$

and covariance matrix Σ^n, these approximate recursive updating equations are

$$\Sigma^{n+1} = \left(\left(\Sigma^n\right)^{-1} + 2\lambda\left(\xi^n\right)\left(x^n\right)\left(x^n\right)^T \right)^{-1}, \tag{8.14}$$

$$\bar{\theta}^{n+1} = \Sigma^{n+1}\left(\left(\Sigma^n\right)^{-1}\bar{\theta}^n + \left(X^{n+1} - \frac{1}{2} \right) x \right). \tag{8.15}$$

The vector $x^n = (1, -p^n)^T$ corresponds to the explanatory variables in (8.4). The function λ is given by

$$\lambda\left(\xi\right) = \frac{\tanh\left(\xi/2\right)}{4\xi}.$$

The value ξ^n is an artificial parameter used in the approximation of the experimental precision. This parameter is also updated recursively, using the equation

$$\xi^{n+1} = \sqrt{\left(x^n\right)^T \Sigma^{n+1} x^n + \left(x^n\right)^T \bar{\theta}^{n+1}}.$$

We can apply the same technique we used for correlated normal beliefs back in Chapter 2 to get a cleaner form for the updating equations

$$\bar{\theta}^{n+1} = \bar{\theta}^n + \frac{\frac{X^{n+1}-\frac{1}{2}}{2\lambda(\xi^n)} - \left(x^n\right)^T\bar{\theta}^n}{\frac{1}{2\lambda(\xi^n)} + \left(x^n\right)^T \Sigma^n x^n} \Sigma^n x^n, \tag{8.16}$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n x^n \left(x^n\right)^T \Sigma^n}{\frac{1}{2\lambda(\xi^n)} + \left(x^n\right)^T \Sigma^n x^n}. \tag{8.17}$$

These equations closely resemble the recursive updating rules we used for linear models in Section 7.2.2. Writing the update in this way gives us some insight into the way the approximation works. The quantity $\frac{1}{2\lambda(\xi)}$ is used in two ways. First, in the denominator of the fractional terms in (8.16) and (8.17), it serves as a stand-in for the variance of the observation. In the correlated normal model considered in Section 2.3.2, the same role is played by the experimental noise. Second, in the numerator of the fractional term in (8.16), the same quantity is used to convert the binary observation $X^{n+1} - \frac{1}{2}$ into a continuous quantity. We can rewrite (8.3) as

$$\mu_1 - \mu_2 p^n = \log\left(\frac{P\left(W^{n+1} \geq p^n\right)}{1 - P\left(W^{n+1} \geq p^n\right)} \right). \tag{8.18}$$

The right-hand side of (8.18) is called the *log-odds* of making a sale. The quantity $\left(x^n\right)^T\bar{\theta}^n$ can be viewed as our prediction of the log-odds. Thus, the continuous quantity $\frac{X^{n+1}-\frac{1}{2}}{2\lambda(\xi^n)}$ can be interpreted as an approximate observation of the log-odds.

We are forcing the problem into the framework of Chapter 7, where our regression model is used to predict continuous observations. To do this, we are artificially creating a set of continuous responses from our binary observations. If $X^{n+1} = 1$, that is, we make the sale, the continuous response is positive, and we conclude that the log-odds are more likely to be greater than zero, and adjust our prior if it is under-estimating them. If we lose the sale and $X^{n+1} = 0$, the continuous response is negative, leading us to believe that the log-odds are more likely to be negative.

One advantage of this approach is that it can easily handle customer attributes, as in (8.4). We simply place a multivariate prior on the vector $\mu = (\mu_1, ..., \mu_{P+1})$ on the parameters of the customer attributes, as well as on the price sensitivity. We then make a pricing decision p^n and apply (8.16) and (8.17) using $x^n = (x_1^n, ..., x_P^n, -p^n)$.

It is important to understand that our posterior distributions in this model are not really normal, just as our posterior distributions in Section 8.1.4.2 were not really Pareto. The updating equations in (8.16) and (8.17) can only serve as an approximation of the way we learn in this problem. The approximation may not always be accurate. We do not necessarily need to use this approach. For example, we can simply collect our observations $X^1, X^2, ..., X^{n+1}$ and fit a logistic regression model to obtain estimates of μ_1 and μ_2. This approach may give us better fits for some particular values of μ, but it is frequentist in nature. It does not incorporate any idea of our prior uncertainty about the parameters, as represented by the prior covariance matrix Σ. Thus, on average across many different truths, we may do better with the Bayesian approximation.

There is no perfect model for learning in this setting, making it difficult to create a sophisticated learning policy. For example, if we try to construct a look-ahead policy such as knowledge gradient, we are forced to rely on an approximation for the predictive distribution of the future beliefs, and another approximation for the optimal implementation decision under those future beliefs. Even so, we can still improve our decision-making by considering some concepts of optimal learning, such as the idea of our uncertainty about the problem parameters.

8.1.6 Bidding strategies

Recall that the revenue function for the logit demand model, under the parameters μ_1 and μ_2, is given by

$$R(p; \mu_1, \mu_2) = \frac{p}{1 + e^{-(\mu_1 - \mu_2 p)}}.$$

Suppose that we have some estimates $\bar{\theta}_1^n$ and $\bar{\theta}_2^n$ of the logistic parameters in (8.3). These estimates may be obtained using the approximate Bayesian model from Section 8.1.5, or they may come from a frequentist statistical procedure. Either way, once we have these estimates, a simple and intuitive course of action would be to simply assume that these are the true values, and make a pricing decision by solving

$$p^n = \arg\max_p R\left(p; \bar{\theta}_1^n, \bar{\theta}_2^n\right) = \arg\max_p \frac{p}{1 + e^{-\left(\bar{\theta}_1^n - \bar{\theta}_2^n p\right)}}. \tag{8.19}$$

For simplicity, let us assume that the set of possible prices is finite. Then, it is very easy to compute (8.19). We need only plug our estimated values into the revenue function and solve. This is known as a *point-estimate* policy, or a *certainty-equivalent* policy. We are making the decision that would be optimal if the true values were exactly equal to our estimates.

8.1.6.1 An idea from multi-armed bandits The main insight of optimal learning, however, is that the true values are *not* exactly equal to our estimates. If we assume that they are, we may under-perform. Consider a very simple bandit problem with a single arm. The one-period reward of the arm follows an exponential distribution with unknown parameter λ. We use a gamma-exponential learning model from Section 2.6.1, and assume that $\lambda \sim Gamma\,(a, b)$.

The average one-period reward is $\frac{1}{\lambda}$, a very simple function of λ. Our estimate of λ is $\mathbb{E}\lambda = \frac{a}{b}$. If we assume that the true value is exactly equal to our estimate, then our resulting estimate of the average one-period reward becomes $\frac{1}{\mathbb{E}\lambda} = \frac{b}{a}$. This reasoning seems straightforward, but it ignores the uncertainty in our beliefs about λ.

Instead of merely plugging in our estimate of λ into the average reward, let us view that reward as a function of a random variable λ. We can then take an expected value of that reward over our distribution of belief, arriving at

$$\mathbb{E}\left(\frac{1}{\lambda}\right) = \frac{b}{a-1}, \tag{8.20}$$

a different estimate of the average reward. This approach accounts for the fact that λ is unknown. Not only do we have an estimate of this parameter, we also have some amount of uncertainty about that estimate. That uncertainty is encoded in the gamma distribution that we use to represent our beliefs.

Suppose now that we have M independent arms. The one-period reward obtained by playing arm x is exponential with parameter λ_x, and we assume that $\lambda_x \sim Gamma\,(a_x, b_x)$. Suppose also that our goal is simply to use a greedy, pure-exploitation strategy for pulling arms. On average over many truth values, the policy that plays $\arg\max_x \frac{b_x}{a_x - 1}$ will do better than the policy that plays $\arg\max_x \frac{b_x}{a_x}$. We do better by incorporating the uncertainty in our beliefs into our decision-making.

8.1.6.2 Bayes-greedy bidding We apply the same idea to the bidding problem. Rather than using (8.19) to make decisions, we take an expectation of the revenue function over our distribution of belief,

$$p^n = \arg\max_p \mathbb{E}R\,(p; \mu_1, \mu_2) = \arg\max_p \mathbb{E}\frac{p}{1 + e^{-(\mu_1 - \mu_2 p)}}. \tag{8.21}$$

Of course, this requires us to have a distribution of belief in the first place. Thus, this policy only really makes sense if we use the Bayesian model from Section 8.1.5 to learn about the unknown parameters. Under this model, we can assume that, at time n, $\mu \sim \mathcal{N}\left(\bar{\theta}^n, \Sigma^n\right)$. After we make a decision p^n and observe X^{n+1}, we use (8.16) and (8.17) to change our beliefs. We refer to this policy as a *Bayes-greedy* policy, to distinguish it from the point-estimate greedy policy.

A well-known property of multivariate normal distributions is that a linear function of a multivariate normal vector is also normal. That is, if $\mu \sim \mathcal{N}\,(\theta, \Sigma)$ and c is a vector, then $c^T x \sim \mathcal{N}\left(c^T\theta, c^T\Sigma c\right)$. In our case, $c = (1, -p)^T$ and

$$\begin{aligned} c^T\theta &= \theta_1 - \theta_2 p, \\ c^T\Sigma c &= \Sigma_{11} - 2p\Sigma_{12} + p^2\Sigma_{22}. \end{aligned}$$

Here we are using the fact that $\Sigma_{12} = \Sigma_{21}$ due to the symmetry of the covariance matrix. Consequently, we can rewrite (8.21) as

$$p^n = \arg\max_p p \cdot \mathbb{E}\left(\frac{1}{1 + e^{-Y}}\right), \tag{8.22}$$

where $Y \sim \mathcal{N}\left(\theta_1 - \theta_2 p, \Sigma_{11} - 2p\Sigma_{12} + p^2\Sigma_{22}\right)$. Note that Y is a one-dimensional normal random variable. The right-hand side of (8.22) now seems straightforward: we need to compute an expectation over a normal density. We can view $\mathbb{E}\left(\frac{1}{1+e^{-Y}}\right)$ as the Bayesian probability of success.

Unfortunately, this expectation is impossible to compute analytically by integration. We have to resort to yet another approximation. One approach is to generate a large number of Monte Carlo samples of Y from a normal distribution with the appropriate mean and variance. We can then take a sample average

$$\mathbb{E}\left(\frac{1}{1+e^{-Y}}\right) \approx \frac{1}{K}\sum_{k=1}^{K} \frac{1}{1+e^{-Y(\omega_k)}}, \tag{8.23}$$

where $Y\left(\omega_k\right)$ is the kth sample realization. Figure 8.6(a) plots the right-hand side of (8.23), as a function of the mean and standard deviation of Y. For very large values of K, we will get more accurate estimates. However, generating enough samples may be fairly expensive computationally.

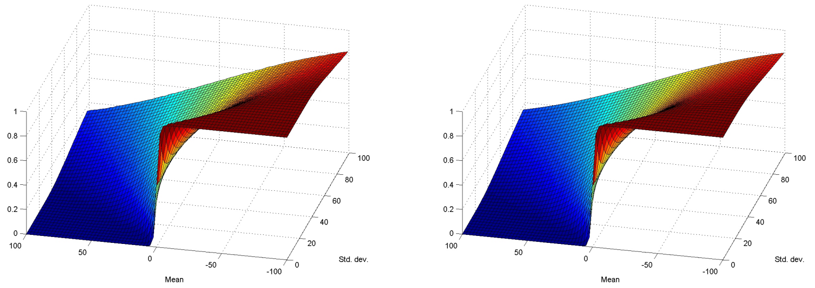
From this figure, we can make an interesting observation. If we fix a value of the standard deviation, and view the left-hand side of (8.23) as a function of the mean of Y, this function also appears to be a logistic curve. In fact, this is borne out by observing that, if $Var\left(Y\right) = 0$, then

$$\mathbb{E}\left(\frac{1}{1+e^{-Y}}\right) = \frac{1}{1+e^{-\mathbb{E}Y}},$$

which is itself a logistic function in $\mathbb{E}Y$. If we fix a larger value of the standard deviation, the expectation continues to look like a logistic function, only with a gentler slope.

This insight has led to a clever closed-form approximation for the difficult expectation. We can write

$$\mathbb{E}\left(\frac{1}{1+e^{-Y}}\right) \approx \frac{1}{1+e^{-\frac{\mathbb{E}Y}{\gamma}}}, \tag{8.24}$$



(a)

(b)

Figure 8.6 Bayesian success probability, as a function of the mean and standard deviation of Y using (a) Monte Carlo simulation, and (b) closed-form approximation.

where

$$\gamma = \sqrt{1 + \frac{\pi^2}{8} Var\left(Y\right)}.$$

Figure 8.6(b) plots this approximation for different values of $\mathbb{E}Y$ and $Var\left(Y\right)$. The result does not exactly match the values we got from Monte Carlo sampling (there are errors on the order of 0.01), but one can easily see that the two surfaces are quite close.

Converting this result back into the language of our bidding problem, our Bayes-greedy policy makes pricing decisions according to the rule

$$p^n = \arg\max_p \frac{p}{1 + e^{-\frac{\bar{\theta}_1^n - \bar{\theta}_2^n p}{\gamma^n(p)}}}, \tag{8.25}$$

where

$$\gamma^n\left(p\right) = \sqrt{1 + \frac{\pi^2}{8}\left(\Sigma_{11}^n - 2p\Sigma_{12}^n + p^2\Sigma_{22}^n\right)}.$$

Interestingly, the calculation in (8.25) is very similar to what we use for the point-estimate policy. The only difference is that we divide the point estimate $\bar{\theta}_1^n - \bar{\theta}_2^n p$ by an additional factor $\gamma^n\left(p\right)$ that depends on the uncertainty in our beliefs (the covariance matrix), as well as the price decision p.

8.1.7 Numerical illustrations

We will examine the performance of Bayes-greedy pricing in an example problem. Suppose that we are running an online bookstore and selling copies of a certain textbook (perhaps this one!) over the Internet. Suppose, furthermore, that the cost of buying the textbook from the publisher wholesale is \$40 per copy. We are thus interested in maximizing the profit function (8.2) with $c = 40$. We will never set a price below \$40, and it is also highly unlikely that anyone will buy the book for more than, say, \$110.

Figure 8.7 shows a realistic starting prior ($\bar{\theta}_1^0 = 15$, $\bar{\theta}_2^0 = 0.2$) for a logistic demand curve in this setting. Notice that the prior logistic curve is a bit narrower than the

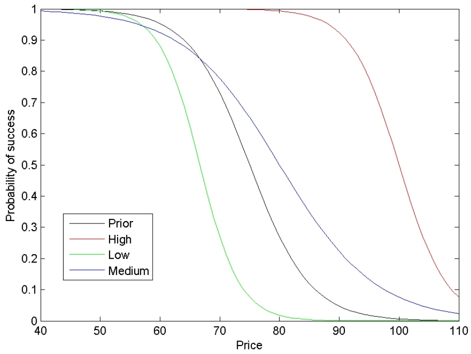
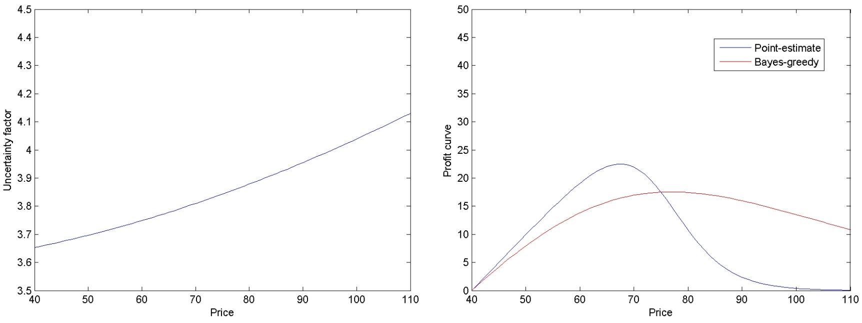


Figure 8.7 Comparison of our starting prior distribution with several possible truth values in a logistic valuation model.



(a) (b)

Figure 8.8 Graphs of (a) the uncertainty factor $\gamma\left(p\right)$ and (b) the modified profit curve as a function of price.

proposed range of \$40–\$110. This prior seems to be saying that we believe customer valuations to be contained roughly in the range $[50, 100]$. However, we also create a prior covariance matrix

$$\Sigma^0 = \begin{bmatrix} 30 & 0 \\ 0 & 0.03^2 \end{bmatrix} \tag{8.26}$$

which allows for some uncertainty in this belief. Note the difference in magnitude between the variances on the market share and price sensitivity. Figure 8.7 also displays three possible true demand curves, representing scenarios where customer valuations are higher, lower, or about the same relative to the prior. The parameters of these truths are as follows:

High truth:	$\mu_1 = 25, \mu_2 = 0.25$
Medium truth:	$\mu_1 = 10, \mu_2 = 0.125$
Low truth:	$\mu_1 = 20, \mu_2 = 0.3$

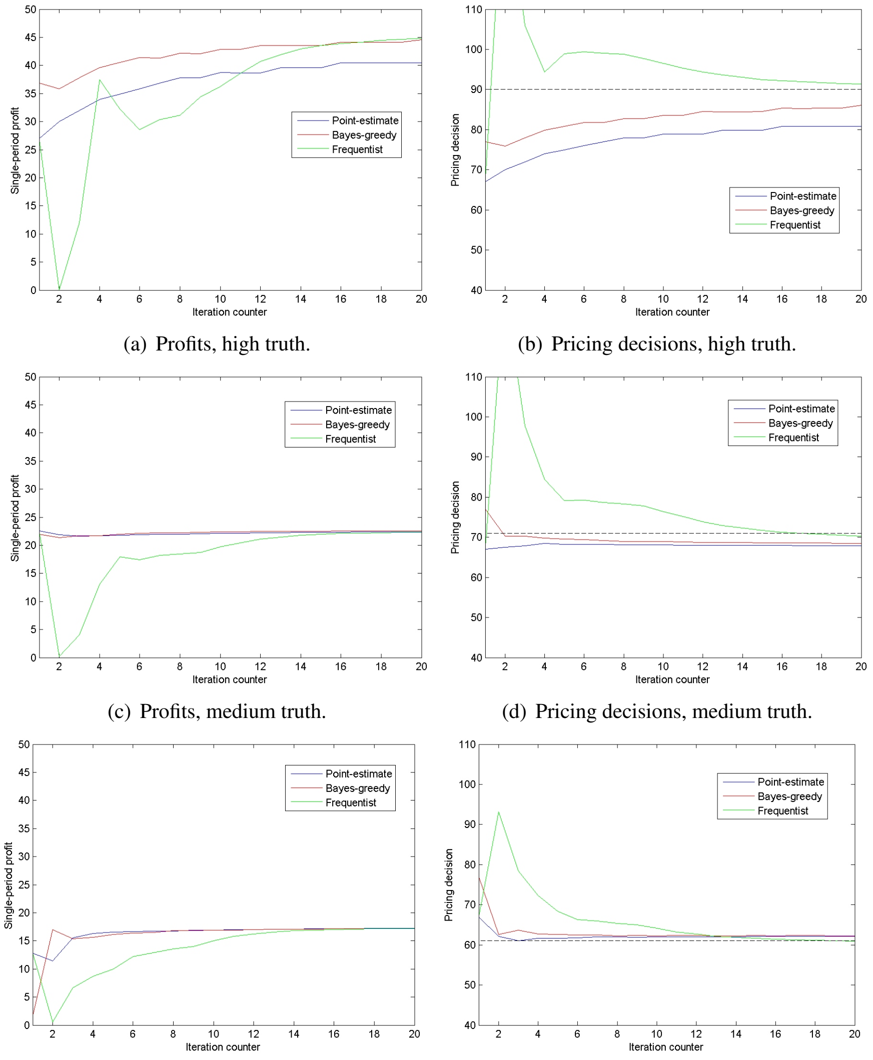
The variation in the price sensitivity for these three scenarios is much smaller than the variation in the market share. This is reflected in our choice of Σ^0. By starting with a prior curve roughly in the middle of our price range, then placing some uncertainty on the parameters, we are able to allow for a wide range of true demand curves.

Figure 8.8(a) shows the uncertainty factor $\gamma^0\left(p\right)$ as a function of the pricing decision p, for the first time step of this problem. We see that $\gamma^0\left(p\right)$ increases with price. We can think of this as expressing the risk involved in choosing higher prices: the penalty for losing the sale is greater, but so is the potential reward if the truth is higher than we think. Figure 8.8(b) compares the estimated profit curves

$$P^{PE}\left(p;\bar{\theta}_1^0,\bar{\theta}_2^0\right) = \frac{p-c}{1+e^{-\left(\bar{\theta}_1^0-\bar{\theta}_2^0 p\right)}}, \tag{8.27}$$

$$P^{BG}\left(p;\bar{\theta}_1^0,\bar{\theta}_2^0\right) = \frac{p-c}{1+e^{-\frac{\bar{\theta}_1^0-\bar{\theta}_2^0 p}{\gamma^0(p)}}}, \tag{8.28}$$

based on the point estimate and the Bayesian distribution, respectively. These are the functions maximized by the point-estimate and Bayes-greedy policies. We see that the Bayes-greedy curve is wider, representing higher uncertainty or variation in the



(a) Profits, high truth.

(b) Pricing decisions, high truth.

(c) Profits, medium truth.

(d) Pricing decisions, medium truth.

(e) Profits, low truth.

(f) Pricing decisions, low truth

Figure 8.9 Average profits and pricing decisions for different policies across 20 iterations.

profit, and the maximum is shifted right. In other words, the Bayes-greedy policy places more value on exploratory behavior (setting higher prices), and makes more aggressive pricing decisions than the point-estimate policy.

It remains to show how this aggressive pricing impacts performance. Figure 8.9 compares the performance (profit, averaged over 1000 sample paths) and pricing decisions of the point-estimate and Bayes-greedy policies in the first 20 iterations, on each of the three truths graphed in Figure 8.7. We also compare these policies to a frequentist logistic regression technique, a standard approach for fitting a logistic distribution to binary observations. We have focused on Bayesian models and algorithms in most earlier chapters. In this setting, however, the Bayesian modeling assumptions do not hold, and our updating equations are approximations. Thus, it is relevant to compare to a frequentist technique in order to see whether the inaccuracy of the Bayesian model has any negative impact on performance. We deliberately use the same set of axes for each comparison, to get a better sense of the magnitude of the difference between policies.

The graphs reveal several interesting behaviors. First, the Bayes-greedy policy consistently prices more aggressively than the point-estimate policy, particularly in the very early iterations. If these aggressive decisions are unsuccessful (for example, if the truth is lower than we believe), the policy quickly corrects its behavior and chooses lower prices. If the optimal price (shown as a dashed line) is higher than expected, both point-estimate and Bayes-greedy gradually ramp up their pricing decisions. However, because Bayes-greedy starts out more aggressively, it is able to make a bigger profit more quickly. Thus, the Bayes-greedy policy exhibits a kind of robustness. If the truth is higher than expected, Bayes-greedy adds considerable value (roughly an extra $10 profit per iteration in Figure 8.9(a)), and if the truth is lower than expected, Bayes-greedy tends to lose the first sale, but adjusts almost immediately and obtains comparable profits to the point-estimate policy in subsequent iterations.

The frequentist method is able to fit a good model eventually. After around 20 iterations, it comes closer to the optimal pricing decision than the two policies using the approximate Bayesian model. At the same time, it does not use any prior information, and so it requires some time in order to fit a good model. In the first ten iterations, it exhibits volatile behavior and consistently chooses prices that are unreasonably high, even in the case when the truth itself is higher than we think. It would seem that Bayes-greedy is able to achieve robust performance in the critical early stages of the problem. In some applications, such as the setting of pricing industrial contracts, ten or twenty iterations may be all we ever get.

8.2 A SAMPLED REPRESENTATION

A powerful approach for handling nonlinear models is to use a sampled representation of the different possible values of the parameter vector θ. When we worked with lookup table representations or linear parametric models, we treated the unknown parameters as following a multivariate normal distribution. While this is a fairly complex representation, the linear structure made it possible to derive simple and elegant updating formulas.

With nonlinear models, it is much harder to develop recursive equations if we were to characterize the unknown parameter vector θ as multivariate normal. We might also note that the multivariate normal distribution can cause difficulties since it allows for virtually every possible value for θ, which can produce unexpected (and inappropriate) behaviors. For example, we might use a quadratic function to approximate a concave surface, but if a sampled value of the coefficient of the quadratic term comes out positive, the curve flips around to a convex surface.

8.2.1 The belief model

We start by assuming that we have a model $f(x|W,\theta)$ that has a known structure, characterized by unknown parameters θ and an exogenous source of experimental noise W, that combines to produce a noisy estimate

$$\hat{y} = f(x|W,\theta),$$

where both W and θ are random variables, although they behave differently. The random variable W changes each time we observe our process. The parameter θ, however, is fixed, but it is fixed at a value that we do not know.

We might think of W as an additive error, as in

$$\hat{y} = f(x|\theta) + W.$$

However, W in $f(x|W,\theta)$ can represent uncertain parameters in a nonlinear function.

We assume that $f(x|W,\theta)$ is nonlinear in θ, which is going to cause us some problems when we need to take expectations as we do in the knowledge gradient. There is a simple way to handle a nonlinear model, which is to assume that θ can only take on one of a sampled set $\hat{\Omega} = \{\theta_1,\ldots,\theta_K\}$. We let our belief state about θ be the set of probabilities $S^n = (p_k^n)_{k=1}^K$ where

$$p_k^n = Prob[\theta = \theta_k|H^n].$$

The variable H^n is called the history of the process, which is given by

$$H^n = (S^0, W^1, \ldots, W^n)$$

A natural way to initialize the probabilities is to start with a uniform prior where

$$p_k^0 = \frac{1}{|\hat{\Omega}|},$$

for all k. For example, if we have 20 samples in $\hat{\Omega}$, we would set $p_k^0 = .05$ for each $k = 1,\ldots,20$. We then let the results of experiments quickly update these probabilities, giving us our belief state

$$S^n = (p_1^n,\ldots,p_K^n).$$

Some thought and care has to be put into how the sample $\theta_1,\ldots,\theta_K$ is created so that we create a reasonable approximation of the distribution of different models that might arise. There are three strategies we might use to generate a sample $\theta_1,\ldots,\theta_k$:

Discretized sample If θ has one to three dimensions, we can simply discretize each dimension and create a discretized version of the parameter space.

Random sampling We could generate each element of θ from a pre-specified range. The problem is that we generally will not know the natural range for each element of θ, especially in the context of a nonlinear, parametric model.

Fitting with prior data Assume we have a set of inputs $x^0, x^1, \ldots, x^n$ with corresponding responses $y^1, y^2, \ldots, y^{n+1}$. We can take a series of subsamples (say, half of the dataset), and then fit a model to this data, producing an estimate θ^k. Repeating this K times provides an initial set of possible values of θ, which automatically captures the correlations between the elements. We suggest using this information to create a multivariate normal distribution, which allows us to scale the covariance matrix if we feel there is not enough variability in the sample.

Fitting with synthetic data It will generally be easier to specify the range over which each dimension of x might vary, as well as the response y, since these have well defined units derived from the contextual setting. We can specify a range for each dimension of x, and then generate random samples (treating each element randomly). We then have to devise a way to generate the responses y, since these are clearly not independent of x. However, we might assume they are independent, since our only goal is to generate a range of $\theta's$ from which to sample. The results of real experiments will quickly fix this.

8.2.2 The experimental model

The next step is to design the experimental model that determines the probability of the response $\hat{y} \sim f^y(y|x, \theta)$. This typically comes from the nature of the problem. Some examples include

- Normal distribution - This is a natural choice when the responses are continuous, and where the noise from an observation of $\hat{y}$ is due to experimental variability, or a range of complex factors. Keep in mind that the normal allows negative outcomes, which may not be appropriate for many situations. However, it is often the case that a normal approximation of the error is very accurate.
- Poisson distribution - This is appropriate when the response represents a count, such as the number of sales of a product, or the number of ad-clicks for an online ad.
- Logistic distribution - This would be appropriate when the response is a 0/1 outcome such as a success or failure.
- Exponential distribution - Exponential distributions are often a good fit when we are observing a time interval, especially when the interval might be quite small. For example, we might be modeling the time between the arrival of patients with a particular condition.

For each distribution, θ captures the parameters of the distribution, such as the mean and variance for a normal distribution, the mean λ

The point is that the distribution $f^y(y|\theta)$ is determined by the characteristics of the problem, guided by a (presumably) unknown parameter vector θ.

8.2.3 Sampling policies

With our sampled belief model,

8.2.4 Resampling

8.3 THE KNOWLEDGE GRADIENT FOR SAMPLED BELIEF MODEL

We are going to demonstrate a powerful method for calculating the knowledge gradient that works with very general nonlinear parametric models. We first return to the knowledge gradient formula, which we repeat here for convenience:

$$\nu^{KG,n}(x) \quad = \quad \mathbb{E}_\theta \mathbb{E}_{W|\theta}\{\max_{x'} f(x',\theta^{n+1}(x))|S^n, x = x^n\} - \mathbb{E}_\theta\{\max_{x'} f(x',\theta)|S^n\}. \tag{8.29}$$

It is useful to break down what is involved in computing the expectations. The first and most complex expectation is over the unknown parameter vector θ which is typically multidimensional, with as few as one or two but perhaps hundreds of dimensions. This same expectation appears in the second term. The second expectation is over the outcome of the experiment, which is a scalar, and possibly binomial (success/failure).

We are going to demonstrate a powerful and practical way for approximating the knowledge gradient by using the sampled belief model, which we first saw in section 2.4. We begin by assuming that the true θ may take on one of a finite (and not too large) set $\theta_1, \ldots, \theta_K$. Let $p^n_k = \mathbb{P}[\theta = \theta^k]$ be the probability that θ_k is the true value, given what we know after n experiments. Finally, we let our belief state $S^n = (\theta_k, p^n_k)^K_{k=1}$.

Using the sampled belief model allows us to compute the conditional expectation given our belief S^n using

$$\begin{aligned}\mathbb{E}_\theta\{\max_{x'} f(x',\theta)|S^n\} &= \bar{f}^n(x') \\ &= \sum_{k=1}^K p^n_k f(x',\theta_k).\end{aligned}$$

Now assume we run an experiment using settings $x^n = x$, and observe the outcome W^{n+1} from the $n+1st$ experiment. Assume that the distribution of W^{n+1} is given by

$$f^W(w|x,\theta) \quad = \quad \mathbb{P}[W^{n+1} = w|x,\theta].$$

We then use this to update our probabilities using the Bayesian updating logic we presented in 2.4, which is given by

$$p^{n+1}_k(x = x^n|W^{n+1} = w) \quad = \quad \frac{1}{C_w} f^W(W^{n+1} = w|x^n,\theta_k)p^n_k, \tag{8.30}$$

where C_w is the normalizing constant given $W^{n+1} = w$, which is calculated using

$$C_w = \sum_{k=1}^K f^W(W^{n+1} = w|x^n,\theta_k)p^n_k.$$

Below, we are going to treat C_W (with capital W) as a random variable with realization C_w.

Using the posterior distribution of belief allows us to write $f(x', \theta^{n+1}(x))$ given S^n using

$$f(x', \theta^{n+1}(x)) = \sum_{k=1}^{K} p_k^{n+1}(x) f(x', \theta_k).$$

Substituting this into equation (8.29) gives us

$$\nu^{KG,n}(x) = \mathbb{E}\left\{\max_{x'} \sum_{k=1}^{K} p_k^{n+1}(x) f(x', \theta_k) | S^n, x = x^n\right\} - \sum_{k=1}^{K} p_k^n f(x, \theta_k). \quad (8.31)$$

Substituting $p_k^{n+1}(x = x^n|W)$ from equation (8.30) into (8.31) gives us

$$\begin{aligned}\nu^{KG,n}(x) = \mathbb{E}_\theta \mathbb{E}_{W|\theta} &\left\{\max_{x'} \frac{1}{C_W} \sum_{k=1}^{K} \left(f^W(W|x^n, \theta_k) p_k^n\right) f(x', \theta_k) | S^n, x = x^n\right\} \\ &- \sum_{k=1}^{K} p_k^n f(x, \theta_k). \qquad (8.32)\end{aligned}$$

We now have to deal with the two outer expectations. The inner expectation $\mathbb{E}_{W|\theta}$ is not too bad since the outcome of an experiment W is a scalar which might be normally distributed or binary (success/failure). The more difficult problem is the outer expectation $\mathbb{E}_\theta$ since θ is often a vector, but here is where we again use our sampled representation of θ.

Keeping in mind that the entire expression is a function of x, the expectation can be written

$$\begin{aligned}&\mathbb{E}_\theta \mathbb{E}_{W|\theta} \left\{\max_{x'} \frac{1}{C_W} \sum_{k=1}^{K} p_k^n f^W(W|x^n, \theta_k) f(x', \theta_k) | S^n, x = x^n\right\} \\ = \quad & \mathbb{E}_\theta \mathbb{E}_{W|\theta} \frac{1}{C_W} \left\{\max_{x'} \sum_{k=1}^{K} p_k^n f^W(W|x^n, \theta_k) f(x', \theta_k) | S^n, x = x^n\right\} \\ = \quad & \sum_{j=1}^{K} \left(\sum_{\ell=1}^{L} \frac{1}{C_{w_\ell}} \left\{\max_{x'} \sum_{k=1}^{K} p_k^n f^W(W = w_\ell|x^n, \theta_k) f(x', \theta_k) | S^n, x = x^n\right\} f^W(W = w_\ell|x, \theta_j)\right) p_j^n. \\ & \qquad (8.33)\end{aligned}$$

If we had to stop here, we would find that equation (8.33) is computationally difficult to compute. The biggest problem is that there are three sums over the set of K values of θ: the outer sum over truths, the inner sum to compute the posterior distribution, and the sum to compute the denominator of the posterior distribution. We then also have the sum over outcomes of W, and the maximization over x'. Fortunately, we can simplify this.

We start by noticing that the terms $f^W(W = w_\ell|x, \theta_j)$ and p_j^n are not a function of x' or k, which means we can take them outside of the $\max$ operator. We can also reverse the order of the other sums over k and w_ℓ, giving us

$$\begin{aligned}\mathbb{E}_\theta\mathbb{E}_{W|\theta}&\left\{\max_{x'}\frac{1}{C_W}\sum_{k=1}^{K}p_k^n f^W(W|x^n,\theta_k)f(x',\theta_k)|S^n,x=x^n\right\}\\ =&\sum_{\ell=1}^{L}\sum_{j=1}^{K}\left(\frac{f^W(W=w_\ell|x,\theta_j)p_j^n}{C_{w_\ell}}\right)\left\{\max_{x'}\sum_{k=1}^{K}p_k^n f^W(W=w_\ell|x^n,\theta_k)f(x',\theta_k)|S^n,x=x^n\right\}.\end{aligned} \tag{8.34}$$

Using the definition of the normalizing constant C_w we can write

$$\begin{aligned}\sum_{j=1}^{K}\left(\frac{f^W(W=w_\ell|x,\theta_j)p_j^n}{C_{w_\ell}}\right) &= \left(\frac{\sum_{j=1}^{K}f^W(W=w_\ell|x,\theta_j)p_j^n}{C_{w_\ell}}\right)\\ &= \left(\frac{\sum_{j=1}^{K}f^W(W=w_\ell|x,\theta_j)p_j^n}{\sum_{k=1}^{K}f^W(W=w_\ell|x,\theta_k)p_k^n}\right)\\ &= 1.\end{aligned}$$

We just simplified the problem by cancelling two summations over the K values of θ. This is a significant simplification, since these sums were nested. This allows us to write (8.34) as

$$\begin{aligned}\mathbb{E}_\theta\mathbb{E}_{W|\theta}&\left\{\max_{x'}\frac{1}{C_W}\sum_{k=1}^{K}p_k^n f^W(W|x^n,\theta_k)f(x',\theta_k)|S^n,x=x^n\right\}\\ &=\sum_{\ell=1}^{L}\left\{\max_{x'}\sum_{k=1}^{K}p_k^n f^W(W=w_\ell|x^n,\theta_k)f(x',\theta_k)|S^n,x=x^n\right\}.\end{aligned} \tag{8.35}$$

Equation (8.35) means that we can compute fairly good approximations of the knowledge gradient even in the presence of a nonlinear belief model, although some care would have to be used if θ was high dimensional.

8.4 LOCALLY QUADRATIC APPROXIMATIONS

8.5 BIBLIOGRAPHIC NOTES

Section 7.2 -

PROBLEMS

8.1 Imagine that we are using a logistics curve to model the probability that a customer will accept a bid, which gives us the revenue function

$$R(p;\theta_1,\theta_2)=\frac{p}{1+e^{-(\theta_1-\theta_2 p)}}. \tag{8.36}$$

Assume the prior is $\bar{\theta}^0=(4,2)$ for prices that are between 0 and 5.

a) Discretize prices into increments of 0.10 and find the optimal price $p^{Exp,1}$ given this prior. This represents the choice you would make using a pure exploitation policy.

b) Assume we choose a price $p^0 = 2$ and we observe that the customer accepts the bid (that is, $W^1 = 1$). Use equations (8.16) and (8.17) to find an updated model. Note that we did not need to assume a variance for the experimental noise. Why is this?

c) Now use the Bayes-greedy policy in equation (8.25) to compute the price $p^{BG,1}$. Try to explain intuitively the difference between $p^{BG,1}$ and $p^{Exp,1}$.

d) Simulate the Bayes-greedy policy for 20 experiments, using (8.16) and (8.17) to update your beliefs. At each iteration, using the beliefs generated by the Bayes-greedy policy, compute $p^{Exp,1}$, and compare the two policies over 20 iterations.

e) Finally, simulate the pure exploitation policy and the Bayes-greedy policy for 20 iterations. Repeat this process 100 times and summarize the difference.