CHAPTER 12

ACTIVE LEARNING IN STATISTICS

We have focused our attention on problems where we are learning in order to make a better decision in some sort of optimization problem, which gives us an economic value of information. It is often the case, however, that we are just trying to fit a statistical model which might be used for a variety of decision problems. We may not know how the model may be used; we are simply interested in carefully choosing what to measure to get the best model.

Most applications of statistics involve problems where we may have to fit a model from a fixed set of observations. This is often referred to as batch statistics. We may also have a process where observations arrive over time, but where we do not have any control over what we observe. This is known as *passive learning*.

There are many situations where we can control the inputs of a process. For example, we may be able to set the price of a product and then observe sales. We may set user preferences on Netflix, thus affecting the movies that are displayed to us; we then choose a movie to rent, which in turn affects what Netflix observes. After we choose the inputs (these might be referred to as independent variables or covariates) x^n, we then observe the response y^{n+1} which is then used to fit the parameters of a model. The machine learning community refers to this process as *active learning*.

In statistics, active learning refers to the ability to control the choice of independent variables. In Chapter 1, we made a distinction between the broad umbrella of active learning, where we make choices with the intent to learn, and the subset of policies

Optimal Learning. By Warren B. Powell and Ilya O. Ryzhov

we call optimal learning, where our choice is guided by a well-defined objective function. We retain this distinction in this chapter, but cover a variety of heuristic and optimal policies for deciding what observations to make when fitting a function. We divide these methods into the following classes:

1) Deterministic policies - These determine all the points to observe in advance, without the benefit of learning the results of any experiments. We consider a special case where these policies are optimal.

2) Heuristic sequential policies - These are active learning policies where the choice of what to observe uses a rule that is not based on any particular performance metric.

3) Variance minimizing policies - These are sequential policies designed to minimize the variance of an estimator.

12.1 DETERMINISTIC POLICIES

There is an extensive literature in statistics that goes under the name design of experiments. The problem is to choose a set of independent variables $x^1, \ldots, x^n$, where x^m is an F-dimensional vector of features which generates an observation $\hat{y}^m$. Let x^n be the vector of independent variables (features), given by

$$x^n = \begin{pmatrix} x_1^n \\ x_1^n \\ \vdots \\ x_F^n \end{pmatrix},$$

where $|\mathcal{F}|$ is a set of features, with $F = |\mathcal{F}|$. Our goal is to fit a linear model

$$y = \sum_{f \in \mathcal{F}} \theta_f x_f + \varepsilon,$$

where ε is an error term.

We want to choose θ to minimize the total errors squared given by

$$\min_{\theta} F(\theta) = \sum_{m=1}^{n} \left(y^m - \sum_{f \in \mathcal{F}} \theta_f x_f^m \right)^2. \tag{12.1}$$

To find the optimal solution, we begin by defining the matrix X^n as

$$X^n = \begin{pmatrix} x_1^1 & x_2^1 & & x_F^1 \\ x_1^2 & x_2^2 & & x_F^2 \\ \vdots & \vdots & \cdots & \vdots \\ x_1^n & x_2^n & & x_F^n \end{pmatrix}.$$

The vector of observations $\hat{y}^1, \ldots, \hat{y}^n$ is represented using

$$Y^n = \begin{pmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^n \end{pmatrix}.$$

As in Chapter 7, the vector θ^n that solves (12.1) is given by

$$\theta^n = [(X^n)^T X^n]^{-1}(X^n)^T Y^n. \tag{12.2}$$

Equation (12.2) gives us an easy way to find the variance of our estimate θ^n. Let v be an n-dimensional random vector, let A be a $F \times n$ deterministic matrix, and let $u = Av$. Let $Cov(v)$ be the covariance matrix of v. Then we can use the identity that

$$Cov(u) = ACov(w)A^T$$

where $Cov(w)$ is the covariance matrix of w. Recall that for matrices A and B, $AB^T = (BA^T)^T$, and that $[(X^n)^T X^n]^{-1}$ is a symmetric matrix. We can use these observations to find the covariance matrix of θ^n if we let $A = [(X^n)^T X^n]^{-1}(X^n)^T$, giving us

$$\begin{aligned} Cov(\theta^n) &= [(X^n)^T X^n]^{-1}(X^n)^T Cov(Y^n)\left([(X^n)^T X^n]^{-1}(X^n)^T\right)^T \\ &= [(X^n)^T X^n]^{-1}(X^n)^T Cov(Y^n)(X^n)[(X^n)^T X^n]^{-1}. \end{aligned}$$

Since the elements of Y^n are independent, $Cov(Y^n) = \sigma_\epsilon^2 I$ where I is the identity matrix and σ_ϵ^2 is the variance of our experimental error . This allows us to write

$$\begin{aligned} Cov(\theta^n) &= [(X^n)^T X^n]^{-1}(X^n)^T X^n [(X^n)^T X^n]^{-1}\sigma_\epsilon^2 \\ &= [(X^n)^T X^n]^{-1}\sigma_\epsilon^2. \end{aligned}$$

The observation error σ_ϵ^2 is fixed; we cannot change it through our choice of x^n. However, we have direct control over the covariance of θ^n by our choice of X^n. Furthermore, we quickly see that the matrix $Cov(\theta^n)$ is a deterministic function of X^n, which means we do not gain anything by observing $\hat{y}^n$. It is for this reason that we can determine the precision of θ^n without making any observations. This is the theoretical foundation of deterministic policies.

So this leaves the question, what do we want to minimize? θ^n is a vector, and $Cov(\theta^n)$ is a matrix. While we would like to minimize all the elements of $Cov(\theta^n)$, we have to choose a single metric. This issue has created a variety of metrics that have come to be known as alphabet-optimality, for reasons that will become clear shortly. A sample of these metrics are

A-optimality - Minimize the average (or trace, which is the sum) of the diagonal elements of $[(X^n)^T X^n]^{-1}$. This is the same as minimizing the average of the variances of each element of θ^n.

C-optimality - Given a weighting vector c, minimize $c^T[(X^n)^T X^n]c$.

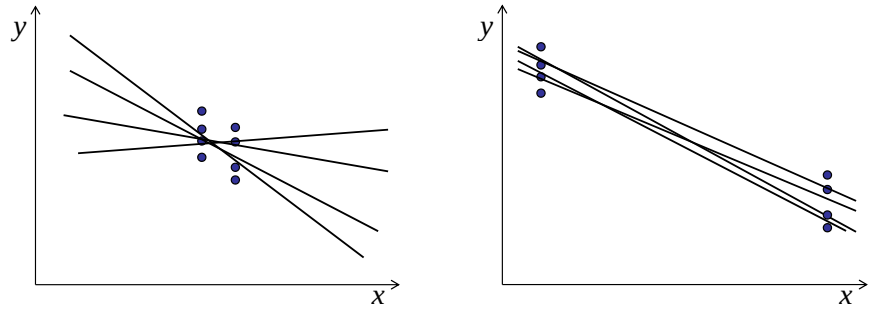

Figure 12.1 Learning a line with closely spaced experiments (a) and experiments with more separation (b).

D-optimality - Minimize $|[(X^n)^T X^n]^{-1}|$, or, equivalently, maximize the determinant of $[(X^n)^T X^n]$.

E-optimality - Maximize the minimum eigenvalue of $[(X^n)^T X^n]$.

G-optimality - Maximize the largest element in the diagonal of $X^n[(X^n)^T X^n](X^n)^T$, which has the effect of minimizing the largest variance of θ^n.

I-optimality - Minimize the average prediction variance over a particular region.

T-optimality - Maximize the trace of $[(X^n)^T X^n]$.

V-optimality - Minimize the average prediction variance over a specific set of points.

All of these methods aim at making the matrix $[(X^n)^T X^n]^{-1}$ smaller in some way, or, equivalently, making the matrix $[(X^n)^T X^n]$ bigger.

We can illustrate the central intuition behind these strategies using the simple example of fitting a line. Figure 12.1(a), reprised here for convenient reading from Figure 7.9, shows the lines that we might estimate if we make a small number of observations that are close to each other. Figure 12.1(b) illustrates the estimates that we might obtain if the experiments are spaced farther apart. It is not surprising that the more widely spaced observations provide a better estimate.

We can quantify this intuitive behavior by computing the matrix $[(X^n)^T X^n]$. Assume we make two observations each at two different locations. An observation x^n might be $(1, 5)$, where the 1 corresponds to the constant term, and the 5 is the value we are measuring. The closely spaced points might be

$$X^n = \begin{pmatrix} 1 & 5 \\ 1 & 5 \\ 1 & 6 \\ 1 & 6 \end{pmatrix}. \tag{12.3}$$

The matrix $[(X^n)^T X^n]$ is given by

$$[(X^n)^T X^n] = \begin{bmatrix} 4 & 22 \\ 22 & 122 \end{bmatrix}.$$

Now assume we measure at more extreme points (but where the average is still the same), given by

$$X^n = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 10 \\ 1 & 10 \end{pmatrix}.$$

The matrix $[(X^n)^T X^n]$ is given by

$$[(X^n)^T X^n] = \begin{bmatrix} 4 & 22 \\ 22 & 202 \end{bmatrix}.$$

The trace of the matrix $[(X^n)^T X^n]$ for the closely spaced points is $4 + 122 = 126$, while the trace for the points that are more spread out is 206. We would say that the second matrix has more "information," and the result is an estimate of θ with lower variance.

It is important that the data be scaled so that the average value of the measured values x remains the same (as we did above). The best way to do this is to simply average the outcomes of the experiments, and compute a corrected value $\bar{X}^n$ for each dimension, using

$$\bar{X}_i^n = X_i^n - \frac{1}{n}\sum_{m=1}^{n} X_i^m.$$

We then compute the matrix using $[(\bar{X}^n)^T \bar{X}^n]$. If we do this to the experimental outcomes in equation (12.3), we would obtain

$$\bar{X}^n = \begin{pmatrix} 0 & -0.5 \\ 0 & -0.5 \\ 0 & 0.5 \\ 0 & 0.5 \end{pmatrix}. \tag{12.4}$$

Regardless of which type of optimality we use, we now have a metric that we can use to help decide which experiments to run. Given a set of potential experiments, we generally want to choose experiments that are distributed around a center, but as far from the center as possible.

It is important to realize that we can choose the best set of potential experiments before making any observations. This property is a unique byproduct of the property that all statistics relating to the reliability of our estimates of θ are purely a function of the experiments and not the observations. We have not enjoyed this property in our previous applications.

12.2 SEQUENTIAL POLICIES FOR CLASSIFICATION

An important class of problems in machine learning is known as classification problems. In this problem class, we are given a set of features x^m for the mth object (this

might be a document, an email or a website), and we are asked to place this document into one of a set of discrete classifications, such as {dangerous, threatening, suspicious, safe}. Often, collecting information about the classification of a document for training purposes is expensive. For example, we may need to ask a security expert to assess the threat level of an email or website. However, given the sheer volume of these sources, we cannot ask a trained expert to provide this information on a large number of documents.

This section describes a series of primarily heuristic search policies for efficiently collecting information for classification problems.

12.2.1 Uncertainty sampling

Let $\hat{y}$ be a discrete quantity that indicates the classification of a document with attributes described by x, and let $P_\theta(\hat{y}|x)$ be the probability of the particular classification $\hat{y}$ if we observe x, given a parameter vector θ. Our goal is to observe the document where our prediction has the highest level of uncertainty. If the classification is binary (for example, dangerous or not), then we want to sample the documents where $P_\theta(\hat{y}|x)$ is as close as possible to 0.5.

If there are more than two outcomes, an alternative strategy is to choose to query the document whose prediction offers the lowest level of confidence, which we compute using

$$x^{LC} = \arg\max_x \left(1 - P_\theta(\hat{y}|x)\right)$$

where

$$\hat{y} = \arg\max_y P_\theta(y|x)$$

is the most likely classification. The idea here is that if the most likely classification in a set is small, then this represents a document (more precisely, a class of documents) where we have a lot of uncertainty, and would benefit from more information. We are trying to mitigate the effect of a worst-case scenario by learning about the document for which we are most likely to make an error in prediction.

This strategy focuses on the most probable classification, and as a result ignores the probabilities of other classifications. For example, we may be much more certain about one classification, implying that more testing is unlikely to change the classification. An alternative strategy is to focus on documents whose most likely classification is close to the second most likely classification. For fixed θ, let

$$\hat{y}_1 = \arg\max_y P_\theta(y|x)$$

and

$$\hat{y}_2 = \arg\max_{y \neq \hat{y}_1} P_\theta(y|x)$$

be the most likely and second most likely classification for a document. The marginal sampling policy chooses the document x where the two highest classification proba-

bilities are the closest. We state this policy using

$$x^M = \arg\min_x \big(P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)\big).$$

The previous two policies look at the level of uncertainty indicated by the most certain document or the difference between the two most certain documents. An idea that takes this thinking a step further uses entropy as a measure of uncertainty. The entropy maximization policy is computed using

$$x^H = \arg\max_x -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

where y_i represents a single classification. Entropy looks at all the potential classifications, and represents an information-theoretic measure of the information required to encode the classification probability distribution.

Empirical research with these policies has produced mixed results. Clearly there will be differences in behavior as the number of potential classifications changes. Not surprisingly, performance depends on the true utility function. The margin and level of confidence policies work better when the goal is to get the classification right, while the entropy maximization policy works if the objective is to minimize the log of the loss from an incorrect classification ("log-loss").

12.2.2 Query by committee

Imagine that we have several competing models for estimating the classification of a document. We would refer to this family of models as a "committee," where $c \in \mathcal{C}$ is a particular model in the set $\mathcal{C}$. Let $P_{\theta^{(c)}}(y_i|x)$ be the probability that model c (parameterized by $\theta^{(c)}$) returns classification y_i given the attributes x of a document.

There are several ways to perform active learning in this setting. One is to let each model vote for a classification. We might record a vote using the indicator function

$$I_c(y_i|x) = \begin{cases} 1 & \text{if } y_i = \arg\max_i P_{\theta^{(c)}}(y_i|x) \\ 0 & \text{otherwise.} \end{cases}$$

The indicator function $I_c(y_i|x)$ simply captures if model c thinks that y_i is the most likely classification. We can then count the number of votes using

$$V(y_i) = \sum_{c \in \mathcal{C}} I_c(y_i|x).$$

Alternatively, we could compute "soft votes" using

$$V(y_i) = \sum_{c \in \mathcal{C}} P_{\theta^{(c)}}(y_i|x).$$

We can then choose to sample the document with the highest *vote entropy*, giving us the policy

$$x^{VE} = \arg\max_x -\sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}.$$

Another measure is the Kullback-Leibler divergence metric, which is a way of measuring the differences between two distributions. We first compute an average probability of classification across the competing models using

$$P_{\mathcal{C}}(y_i|x) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} P_{\theta^{(c)}}(y_i|x).$$

The KL divergence is then computed using

$$D(P_{\theta^{(c)}}\|P_{\mathcal{C}}) = \sum_i P_{\theta^{(c)}}(y_i|x)\log\frac{P_{\theta^{(c)}}(y_i|x)}{P_{\mathcal{C}}(y_i|x)}.$$

The Kullback-Leibler divergence is a type of distance metric since it is measuring the degree of similarity between two probability distributions: the distribution (across potential classifications) obtained when we average all the probabilities across the competing models, versus the distribution produced by each model. The quantity $D(P_{\theta^{(c)}}\|P_{\mathcal{C}})$ is the KL divergence for a particular model c. Our policy for deciding which document to evaluate is obtained by maximizing the average KL divergence across all the models, given by

$$x^{KL} = \arg\max_x \frac{1}{C}\sum_{c\in\mathcal{C}} D(P_{\theta^{(c)}}\|P_{\mathcal{C}}).$$

The idea with this policy is to choose to evaluate the document with the greatest disagreement among the different models. If the competing models largely agree, then it is unlikely that more information will change this consensus. Additional information is likely to have the greatest impact when competing models disagree.

12.2.3 Expected error reduction

An interesting policy uses an estimate of the degree to which information reduces the likelihood of being incorrect. We are going to assume that we have a set of unlabeled documents $\mathcal{U}$, which means that we have not solicited a classification from a domain expert. Assume we are considering the possibility of collecting information on the document x. We do not know its classification, but our current estimate of the probability that it will be classified as y_i is $P_\theta(y_i|x)$. If we choose document x and observe the classification y_i, we can use this information to update our classification probability, which we represent using $P_{\theta^{+(x,y_i)}}(y|x)$. This is analogous to a posterior belief.

Now, we are going to use this updated probability model on each document $u \in \mathcal{U}$, with feature vector x^u. We let

$$\hat{y}^u = \arg\max_y P_\theta(y|x^u)$$

be the most likely classification for a particular document x^u in our set $\mathcal{U}$. If $P_{\theta^{+(x,y_i)}}(\hat{y}^u|x^u)$ is the probability of this most likely classification, then $1 - P_{\theta^{+(x,y_i)}}(\hat{y}^u|x^u)$ represents the probability that the document u does not belong to the class that we think is the most likely. This is analogous to our posterior

belief about how likely we are to make a mistake about document u after observing a result for x. We then choose x to minimize this probability of error in expectation over all possible classifications y_i of document x. The policy, then, is given by

$$x^{ER} = \arg\min_x \sum_i P_\theta(y_i|x) \left(\sum_{u \in \mathcal{U}} \left(1 - P_{\theta^{+(x,y_i)}}(\hat{y}^u|x^u)\right) \right).$$

12.3 A VARIANCE MINIMIZING POLICY

We are going to describe an information collecting policy with the goal of reducing variance, but this time we are going to address a richer set of models and issues than we encountered in Section 12.1 when the goal was to minimize variance measures of the regression vector θ.

We start by assuming that there exists a model $y(x) = f(x) + \epsilon$ where $f(x)$ is the true model and ϵ is a source of experimental noise over which we have no control. Our goal is to collect a training dataset $\mathcal{D}^n = \left\{(x^0, y^1), (x^1, y^2), \ldots, (x^{n-1}, y^n)\right\}$. We group the choice $(x^0, \ldots, x^{n-1})$ of what to measure with the corresponding observations $(y^1, \ldots, y^n)$ into a single training dataset. We note that the standard notation in statistics is to let x^1 be the experiment that produces y^1, but as the development below illustrates, it is cleaner to let the superscript capture the information content; when we choose x^m, y^{m+1} is a random variable.

We assume we are using a sequential policy to determine $x^0, x^1, x^2, \ldots$ (starting with x^0) which may depend on the outcomes $y^1, y^2, \ldots$. This means that $\mathcal{D}^n$ is a random set, constructed sequentially. We use this information to fit an approximation $\bar{y}^n = \bar{f}^n(x|\mathcal{D}^n)$. Below, we are going to write the prediction as $\bar{y}^n(x|\mathcal{D}^n)$ to express its dependence on the query point x, and the data $\mathcal{D}^n$.

The goal is to design a policy that minimizes the variance in our predictions $\bar{y}^n(x|\mathcal{D}^n)$, but this depends on the points x where we *might* query the function. We did not have to deal with this issue in Section 12.1, but now we are going to assume that we are given a distribution $P(x)$ that gives the probability that we will want an estimate of the function at x. In practice, $P(x)$ may be chosen to be uniform over some region, or we may specify a normal distribution with some mean μ_x and a spread σ_x. Alternatively, we may have a sequence of observations $\tilde{x}^1, \ldots, \tilde{x}^k$ from history which can serve as a probability distribution. Either way, $P(x)$ serves as a weighting function that tells us the region in which we are interested. However, it is important to recognize that the probability that x is in the random set $\mathcal{D}^n$, which is influenced by our learning policy, may be completely different than $P(x)$.

If we make an observation at x, we are going to observe

$$y(x) = f(x) + \epsilon,$$

where $f(x)$ is our true (but unknown) function, and ϵ is the inherent noise in the observation. Our approximation is going to give us the estimate $\bar{y}^n(x|\mathcal{D}^n)$. A reasonable goal is to design a policy for choosing an experiment x to minimize the prediction error $(\bar{y}^n(x|\mathcal{D}^n) - y(x))^2$ for a single realization of $y(x)$ (which is random

because of ϵ), and a single estimate $\bar{y}^n(x|\mathcal{D}^n)$ from a dataset $\mathcal{D}^n$ (which is random because $\mathcal{D}^n$ is random). To formalize this idea, we need to take expectations. We let $\mathbb{E}_T(\cdot)$ be the total expectation over the observation noise imbedded in $y(x)$ and the observation of $\mathcal{D}^n$. These are independent, so we can write them as

$$\mathbb{E}_T(\bar{y}^n(x|\mathcal{D}^n) - y(x))^2 = \mathbb{E}_y\mathbb{E}_{\mathcal{D}}(y(x) - \bar{y}^n(x|\mathcal{D}^n))^2, \tag{12.5}$$

where $\mathbb{E}_{\mathcal{D}}$ is the expectation over all the possible outcomes of $\mathcal{D}^n$, and $\mathbb{E}_y$ is the expectation over all possible realizations of $y(x)$. We can break down the expected total variation (for a given x) in (12.5) using

$$\begin{aligned}\mathbb{E}_T(\bar{y}^n(x|\mathcal{D}^n) - y(x))^2 &= \mathbb{E}[(y(x) - \mathbb{E}[y(x)])^2] \\ &\quad + \big(\mathbb{E}_{\mathcal{D}}[\bar{y}(x|\mathcal{D}^n)] - \mathbb{E}[y(x)]\big)^2 \\ &\quad + \mathbb{E}_{\mathcal{D}}\big[(\bar{y}(x|\mathcal{D}^n) - \mathbb{E}_{\mathcal{D}}[\bar{y}(x|\mathcal{D}^n)])^2\big].\end{aligned} \tag{12.6}$$

The first term reflects the pure noise due to ϵ, which will not be affected by the learning policy. The second term captures the bias in the model, which is purely a function of the structural form of the underlying model, as well as the choice of x. Again, this is not affected by the choice of $\mathcal{D}^n$. The right-hand side of equation (12.6) is also known as a *bias-variance decomposition.*

The third term is the variance due to the estimation of the model from the training data. This is where we capture the variation in the estimated model (after n observations) due to the different variations in $\mathcal{D}^n$ that might be produced by following a specific learning policy. The variations in $\mathcal{D}^n$ arise because of differences in realizations in the observations $y^1, \ldots, y^n$ that lead to different decisions $x^2, \ldots, x^n$. This is the term that we wish to minimize by choosing a good learning policy.

For compactness, we are going to let

$$\sigma_y^{2,n}(x) = \mathbb{E}_{\mathcal{D}}\big(\bar{y}^n(x|\mathcal{D}^n) - \mathbb{E}_{\mathcal{D}}[\bar{y}^n(x|\mathcal{D}^n)]\big)^2.$$

Now assume that we are considering the possibility of adding (x^n, y^{n+1}) to create an expanded dataset

$$\mathcal{D}^{n+1} = \mathcal{D}^n \cup (x^n, y^{n+1}).$$

The choice x^n is a deterministic quantity (given $\mathcal{D}^n$) that we are thinking of measuring, while y^{n+1} is the random observation that we have not yet observed when we choose x^n. Let $\mathbb{E}_{\mathcal{D}}^n$ be the conditional expectation given $\mathcal{D}^n$. The choice of observation x^n is a deterministic function of $\mathcal{D}^n$. The new information in $\mathcal{D}^{n+1}$ is y^{n+1}. If we choose x^n, let $\tilde{\sigma}_y^{2,n}(x|x^n)$ be the conditional variance (given $\mathcal{D}^n$) of our estimate if we choose x^n, which is given by

$$\tilde{\sigma}_y^{2,n}(x|x^n) = \mathbb{E}_{\mathcal{D}}^n\big(\bar{y}^{n+1}(x|\mathcal{D}^{n+1}) - \mathbb{E}_{\mathcal{D}}^n[\bar{y}^{n+1}(x|\mathcal{D}^{n+1})]\big)^2. \tag{12.7}$$

The variance $\tilde{\sigma}_y^{2,n}(x|x^n)$ is defined in exactly the same way as $\tilde{\sigma}$ in (2.9). However, in Chapter 2, we also proved that this quantity coincided with the change in variance between two experiments, under a Bayesian learning model. This equivalence does *not* hold in the frequentist model we are considering here. In (12.7), $\tilde{\sigma}^{2,n}$ represents only the conditional variance of our estimate given a choice of experiment, and

minimizing this quantity is equivalent to minimizing the future variance of our prediction (similar to the posterior variance considered in the Bayesian models). If we were to adopt a Bayesian approach here, $\mathbb{E}[y(x)]$ would itself be random (since we do not know the true mean of the observation), and we would need to consider all the terms in (12.6) together in order to minimize the posterior variance.

In the frequentist model, it is enough to minimize the last term in (12.6). We want to choose x^n so as to minimize (12.7). However, we do not know that we are going to want to observe the function at x, so we need to take the expectation over potential observations using our marginal distribution $P(x)$, giving us

$$\bar{\sigma}_y^{2,n}(x^n) = \int_x \tilde{\sigma}_y^{2,n}(x|x^n)P(x)dx. \tag{12.8}$$

The variance $\bar{\sigma}_y^{2,n}(x^n)$ is the expected variance given that we run the experiment x^n. Our policy will be to choose the value x^n that produces the greatest reduction in variance. We state this policy using

$$x^n = \arg\min_{x'} \bar{\sigma}_y^{2,n}(x'). \tag{12.9}$$

The value of this policy is that it is fairly easy to compute. We illustrate the calculations for an approximation architecture that uses mixtures of Gaussians.

12.4 MIXTURES OF GAUSSIANS

In Section 12.1, we considered the case of estimating a single regression function. Here, we turn to a more general approximation architecture where we assume that observations come from one of a set of populations that we index $i = 1, \ldots, N$. We assume that each population produces a different behavior that can be reasonably approximated by a line, as illustrated in Figure 12.2.

12.4.1 Estimating parameters

We need to estimate the mean and variance of the experiments (independent variables) x, given by $\mu_{x,i}$ and $\sigma^2_{x,i}$, the mean and variance of the observations y, given by $\mu_{y,i}$ and $\sigma^2_{y,i}$, and the covariance between x and y, $\sigma_{xy,i}$. Temporarily assume that for each population i, we know the expected value of the observation x, $\mu_{x,i}$, and its variance $\sigma^2_{x,i}$. We are going to begin by presenting calculations for the remaining moments, after which we show how to compute $\bar{\sigma}^2(x^n)$.

We can represent the joint distribution of x and y for a population i. First define

$$z = \begin{bmatrix} x \\ y \end{bmatrix}, \ \mu_i = \begin{bmatrix} \mu_{x,i} \\ \mu_{y,i} \end{bmatrix}, \ \Sigma_i = \begin{bmatrix} \sigma^2_{x,i} & \sigma_{xy,i} \\ \sigma_{xy,i} & \sigma^2_{y,i} \end{bmatrix}.$$

We can now write the joint distribution of x and y for group i using

$$P(x, y|i) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} \exp\left[-\frac{1}{2}(z-\mu_i)^T\Sigma_i^{-1}(z-\mu_i)\right].$$

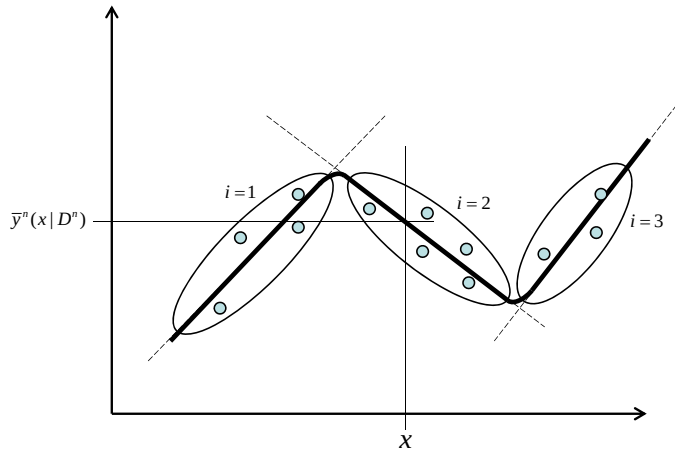

Figure 12.2 Illustration of fitting data using mixtures of Gaussians (based on Cohn et al. 1994).

We will not know the true means and variances, but we can estimate them from an initial sample.

The conditional variance of y given x is given by

$$\sigma^2_{y|x,i} = \sigma^2_{y,i} - \frac{\sigma_{xy,i}}{\sigma^2_{x,i}}.$$

The conditional expectation $\bar{y}^n_i(x|\mathcal{D}^n)$ and variance $\sigma^2_{\hat{y},i}$ given x are

$$\begin{aligned} \bar{y}^n_i(x|\mathcal{D}^n) &= \mu_{y,i}(x) + \frac{\sigma_{xy,i}}{\sigma^2_{x,i}}(x - \mu_{x,i}), \\ \sigma^2_{y,i} &= \frac{\sigma^2_{y|x,i}}{n_i}\left(1 + \frac{(x-\mu_{x,i})^2}{\sigma^2_{x,i}}\right). \end{aligned}$$

The scalar n_i can be viewed as the weight that group i should be given, calculated using

$$n_i = \sum_{j=1}^{m} \frac{P(x_j, y_j|i)}{\sum_{k=1}^{N} P(x_j, y_j|k)}.$$

We next compute the probability that population i contributes to the observation corresponding to x using

$$h_i(x) = \frac{P(x|i)}{\sum_{j=1}^{N} P(x|j)},$$

where

$$P(x|i) = \frac{1}{\sqrt{2\pi\sigma^2_{x,i}}} \exp\left[-\frac{(x-\mu_{x,i})^2}{2\sigma^2_{x,i}}\right]. \tag{12.10}$$

Given the observation x, the expectation across all the populations of $\bar{y}_i^n(x|\mathcal{D}^n)$ and its variance are given by

$$\begin{aligned}\bar{y}^n(x|\mathcal{D}^n) &= \sum_{i=1}^{N} h_i(x)\bar{y}_i^n(x|\mathcal{D}^n),\\ \sigma_y^2(x|\mathcal{D}^n) &= \sum_{i=1}^{N} \frac{h_i^2\sigma_{y|x,i}^2}{n_i}\left(1+\frac{(x-\mu_{x,i})^2}{\sigma_{x,i}^2}\right).\end{aligned}$$

We now have the foundation we need to estimate the expected value of an additional experiment.

12.4.2 Active learning

We have to calculate the variance given an experiment x^n, where we continue to assume that $P(x)$ is known. We have assumed that $\mu_{x,i}$ and $\sigma_{x,i}^2$ are known for each population i, but we cannot derive these statistics directly from the marginal distribution $P(x)$. For example, we may approximate $P(x)$ by assuming that it is uniformly distributed over some region, or we may represent it from a sample obtained from history.

We could try to estimate $\mu_{x,i}$ and $\sigma_{x,i}^2$ from training data, but these observations are chosen according to our learning policy, and may not reflect the true distribution that reflects the likelihood of actually needing to calculate the function at some x. If we use the training sample, we are effectively using the joint distribution $P(x^n, y|i)$ instead of $P(x, y|i)$. We can correct for the difference in the sampling distribution (at least in principle) using

$$P(x, y|i) = P(x^n, y|i)\frac{P(x|i)}{P(x^n|i)}.$$

We compute the conditional distribution $P(x|i)$ by using equation (12.10) with the mean and variance $\mu_{x,i}$ and $\sigma_{x,i}^{2,n}$ computing using data sampled from $P(x)$ or a reference sample. The distribution $P(x^n, y|i)$ is also computed using equation (12.10) with mean and variance calculated from a training dataset.

We now have to compute $\bar{\sigma}^{2,n}(x^n)$ which we use in equation (12.9) to determine the next point to measure. We first calculate $\tilde{\sigma}_y^{2,n}(x|x^n)$ for a particular query point x. The distribution of y^{n+1} given x^n can be calculated as a mixture of normal distributions, given by

$$\begin{aligned}P(y^{n+1}|x^n) &= \sum_{i=1}^{N} h_i(x^n)P(y^{n+1}|x^n, i)\\ &= \sum_{i=1}^{N} h_i(x^n)\mathcal{N}\big(\bar{y}_i^n(x^n|\mathcal{D}^n), \sigma_{y|x,i}^{2,n}(x^n)\big),\end{aligned}$$

where $\mathcal{N}(\mu, \sigma^2)$ represents the normal distribution, where $\bar{y}_i^n(x^n|\mathcal{D}^n)$ is the mean of y^{n+1} based on data for group i, and $\sigma_{y|x,i}^2(x^n)$ is the variance. The conditional

variance of our prediction of x given the choice of x^n can be found using

$$\tilde{\sigma}_y^{2,n}(x|x^n) = \sum_{i=1}^{n} \frac{h_i^2(x)\tilde{\sigma}_{y|x,i}^{2,n}(x|x^{n+1})}{n_i + h_i(x|x^{n+1})} \left(1 + \frac{(x-\mu_{x,i})}{\sigma_{x,i}^2}\right),$$

where

$$\begin{aligned}
\tilde{\sigma}_{y,i}^{2,n}(x|x^{n+1}) &= \frac{n_i\sigma_{y,i}^{2,n}}{n_i + h_i(x|x^{n+1})} \\
&\quad + \frac{n_i + h_i(x|x^{n+1})\left(\sigma_{y|x,i}^2 + (\bar{y}_i^{n+1}(x|x^{n+1}) - \mu_{y,i})^2\right)}{(n_i + h_i(x|x^{n+1}))^2}, \\
\tilde{\sigma}_{y|x,i}^{2,n}(x|x^{n+1}) &= \tilde{\sigma}_{y,i}^{2,n}(x|x^{n+1}) - \frac{\tilde{\sigma}_{xy,i}^{2,n}}{\sigma_{x,i}^2}, \\
\tilde{\sigma}_{xy,i}(x|x^{n+1}) &= \frac{n_i\sigma_{xy,i}}{n_i + h_i(x|x^{n+1})} \\
&\quad + \frac{n_i h_i(x|x^{n+1})(x^{n+1} - \mu_{x,i})(\bar{y}_i^{n+1}(x|x^{n+1}) - \mu_{y,i})}{(n_i + h_i(x|x^{n+1}))^2}, \\
\tilde{\sigma}_{xy,i}^2(x|x^{n+1}) &= (\tilde{\sigma}_{xy,i})^2 + \frac{n_i^2(h_i(x|x^{n+1}))^2\sigma_{y|x^{n+1},i}^2(x^{n+1} - \mu_{x,i})^2}{\cdot}
\end{aligned}$$

If we are estimating $\mu_{x,i}$ and $\sigma_{x,i}^2(x|x^{n+1})$ from data, we can use the following equations

$$\begin{aligned}
\bar{\mu}_{x,i}^{n+1}(x^{n+1}) &= \frac{n_i\bar{\mu}_{x,i}^n + h_i(x|x^{n+1})x^{n+1}}{n_i + h_i(x|x^{n+1})}, \\
\sigma_{x,i}^{2,n+1}(x|x^{n+1}) &= \frac{n_i\sigma_{x,i}^{2,n}}{n_i + h_i(x|x^{n+1})} + \frac{n_i h_i(x|x^{n+1})(x^{n+1} - \bar{\mu}_{x,i}^n)^2}{(n_i + h_i(x|x^{n+1}))^2}.
\end{aligned}$$

With $\tilde{\sigma}_y^{2,n}(x|x^n)$ in hand, we use equation (12.8) to integrate over the different query points x to obtain $\bar{\sigma}_y^{2,n}(x^n)$.

12.5 BIBLIOGRAPHIC NOTES

Section 12.1 - This section covers classic material from experimental design as it is known within the statistics community (see DeGroot 1970, Wetherill & Glazebrook 1986, Montgomery 2008).

Section 12.2 - This section is based on Settles (2009).

Section 12.3 - This section is from Geman et al. (1992), with material from Cohn et al. (1994), Cohn et al. (1996) and Settles (2009). See Hastie et al. (2005) for a nice discussion of bias-variance decomposition.

12.5.1 Optimal computing budget allocation

The value of the indifference zone strategy is that it focuses on achieving a specific level of solution quality, being constrained by a specific budget. However, it is often the case that we are trying to do the best we can within a specific computing budget. For this purpose, a line of research has evolved under the name *optimal computing budget allocation*, or OCBA.

Figure 12.3 illustrates a typical version of an OCBA algorithm. The algorithm proceeds by taking an initial sample $N_x^0 = n_0$ of each alternative $x \in \mathcal{X}$, which means we use $B^0 = Mn_0$ experiments from our budget B. Letting $M = |\mathcal{X}|$, we divide the remaining budget of experiments $B - B^0$ into equal increments of size Δ, so that we do $N = (B - Mn_0)\Delta$ iterations.

After n iterations, assume that we have measured alternative x N_x^n times, and let W_x^m be the mth observation of x, for $m = 1, \ldots, N_x^n$. The updated estimate of the value of each alternative x is given by

$$\theta_x^n = \frac{1}{N_x^n} \sum_{m=1}^{N_x^n} W_x^m.$$

Let $x^n = \arg\max \theta_x^n$ be the current best option.

After using Mn_0 observations from our budget, at each iteration we increase our allowed budget by $B^n = B^{n-1} + \Delta$ until we reach $B^N = B$. After each increment, the allocation $N_x^n,\ x \in \mathcal{X}$ is recomputed using

$$\frac{N_x^{n+1}}{N_{x'}^{n+1}} = \frac{\hat{\sigma}_x^{2,n}/(\theta_{x^n}^n - \theta_{x'}^n)^2}{\hat{\sigma}_{x'}^{2,n}/(\theta_{x^n}^n - \theta_{x'}^n)^2} \quad x \neq x' \neq x^n, \tag{12.11}$$

$$N_{x^n}^{n+1} = \hat{\sigma}_{x^n}^n \sqrt{\sum_{i=1, i\neq x^n}^{M} \left(\frac{N_x^{n+1}}{\hat{\sigma}_i^n}\right)^2}. \tag{12.12}$$

We use equations (12.11)-(12.12) to produce an allocation N_x^n such that $\sum_x N_x^n = B^n$. Note that after increasing the budget, it is not guaranteed that $N_x^n \geq N_x^{n-1}$ for some x. If this is the case, we would not measure these alternatives at all in the next iteration. We can solve these equations by writing each N_x^n in terms of some fixed alternative (other than x^n), such as N_1^n (assuming $x^n \neq 1$). After writing N_x^n as a function of N_1^n for all x, we then determine N_1^n so that $\sum N_x^n \approx B^n$ (within rounding).

The complete algorithm is summarized in figure 12.3.

Step 0. Initialization:

Step 0a. Given a computing budget B, let n^0 be the initial sample size for each of the $M = |\mathcal{X}|$ alternatives. Divide the remaining budget $T - Mn_0$ into increments so that $N = (T - Mn_0)/\delta$ is an integer.

Step 0b. Obtain samples W_x^m, $m = 1, \ldots, n_0$ samples of each $x \in \mathcal{X}$.

Step 0c. Initialize $N_x^1 = n_0$ for all $x \in \mathcal{X}$.

Step 0d. Initialize $n = 1$.

Step 1. Compute

$$\theta_x^n = \frac{1}{N_x^n} \sum_{m=1}^{N_x^n} W_x^m.$$

Compute the sample variances for each pair using

$$\hat{\sigma}_x^{2,n} = \frac{1}{N_x^n - 1} \sum_{m=1}^{N_x^n} (W_x^m - \theta_x^n)^2 .$$

Step 2. Let $x^n = \arg\max_{x \in \mathcal{X}} \theta_x^n$.

Step 3. Increase the computing budget by Δ and calculate the new allocation $N_1^{n+1}, \ldots, N_M^{n+1}$ so that

$$\begin{aligned} \frac{N_x^{n+1}}{N_{x'}^{n+1}} &= \frac{\hat{\sigma}_x^{2,n}/(\theta_{x^n}^n - \theta_{x'}^n)^2}{\hat{\sigma}_{x'}^{2,n}/(\theta_{x^n}^n - \theta_{x'}^n)^2} \quad x \neq x' \neq x^n, \\ N_{x^n}^{n+1} &= \hat{\sigma}_{x^n}^n \sqrt{\sum_{i=1, i \neq x^n}^{M} \left(\frac{N_x^{n+1}}{\hat{\sigma}_i^n} \right)^2}. \end{aligned}$$

Step 4. Perform $\max\left(N_x^{n+1} - N_x^n, 0\right)$ additional simulations for each alternative x.

Step 5. Set $n = n + 1$. If $\sum_{x \in \mathcal{X}} N_x^n < B$, go to step 1.

Step 6. Return $x^n \arg\max_{x \in \mathcal{X}} \theta_x^n$.

Figure 12.3 Optimal computing budget allocation procedure.