14 EM Algorithms

14.1 Introduction

In Chapter 8, we discussed methods for maximizing the log-likelihood (LL) function. As models become more complex, maximization by these methods becomes more difficult. Several issues contribute to the difficulty. First, greater flexibility and realism in a model are usually attained by increasing the number of parameters. However, the procedures in Chapter 8 require that the gradient be calculated with respect to each parameter, which becomes increasingly time consuming as the number of parameters rises. The Hessian, or approximate Hessian, must be calculated and inverted; with a large number of parameters, the inversion can be numerically difficult. Also, as the number of parameters grows, the search for the maximizing values is over a larger-dimensioned space, such that locating the maximum requires more iterations. In short, each iteration takes longer and more iterations are required.

Second, the LL function for simple models is often approximately quadratic, such that the procedures in Chapter 8 operate effectively. As the model becomes more complex, however, the LL function usually becomes less like a quadratic, at least in some regions of the parameter space. This issue can manifest itself in two ways. The iterative procedure can get "stuck" in the nonquadratic areas of the LL function, taking tiny steps without much improvement in the LL. Or the procedure can repeatedly "bounce over" the maximum, taking large steps in each iteration but without being able to locate the maximum.

Another issue arises that is more fundamental than the count of parameters and the shape of the LL function. Usually, a researcher specifies a more general, and hence complex, model because the researcher wants to rely less on assumptions and, instead, obtain more information from the data. However, the goal of obtaining more information from the data is inherently at odds with simplicity of estimation.

Expectation-maximization (EM) algorithms are procedures for maximizing an LL function when standard procedures are numerically

difficult or infeasible. The procedure was introduced by Dempster, Laird, and Rubin (1977) as a way of handling missing data. However, it is applicable far more generally and has been used successfully in many fields of statistics. McLachlan and Krishnan (1997) provide a review of applications. In the field of discrete choice modeling, EM algorithms have been used by Bhat (1997a) and Train (2008a,b).

The procedure consists of defining a particular expectation and then maximizing it (hence the name). This expectation is related to the LL function in a way that we will describe, but it differs in a way that facilitates maximization. The procedure is iterative, starting at some initial value for the parameters and updating the values in each iteration. The updated parameters in each iteration are the values that maximize the expectation in that particular iteration. As we will show, repeated maximization of this function converges to the maximum of LL function itself.

In this chapter, we describe the EM algorithm in general and develop specific algorithms for discrete choice models with random coefficients. We show that the EM algorithm can be used to estimate very flexible distributions of preferences, including nonparametric specifications that can approximate asymptotically any true underlying distribution. We apply the methods in a case study of consumers' choice between hydrogen and gas-powered vehicles.

14.2 General Procedure

In this section we describe the EM procedure in a highly general way so as to elucidate its features. In subsequent sections, we apply the general procedure to specific models. Let the observed dependent variables be denoted collectively as y, representing the choices or sequence of choices for an entire sample of decision makers. The choices depend on observed explanatory variables that, for notational convenience, we do not explicitly denote. The choices also depend on data that are missing, denoted collectively as z. Since the values of these missing data are not observed, the researcher specifies a distribution that represents the values that the missing data could take. For example, if the income of some sampled individuals is missing, the distribution of income in the population can be a useful specification for the distribution of the missing income values. The density of the missing data is denoted $f(x \mid \theta)$, which depends in general on parameters θ to be estimated.

The behavioral model relates the observed and missing data to the choices. This model predicts the choices that would arise if the missing data were actually observed instead of being missing. This behavioral

model is denoted $P(y \mid z, \theta)$, where θ are parameters that may overlap or extend those in f. (For notational compactness, we use θ to denote all the parameters to be estimated, including those entering f and those entering P.) Since, however, the missing data are in fact missing, the probability of the observed choices, using the information that the researcher observes, is the integral of the conditional probability over the density of the missing data:[1]

$$P(y \mid \theta) = \int P(y \mid z, \theta) f(z \mid \theta) dz.$$

The density of the missing data, $f(z \mid \theta)$, is used to predict the observed choices and hence does not depend on y. However, we can obtain some information about the missing data by observing the choices that were made. For example, in vehicle choice, if a person's income is missing but the person is observed to have bought a Mercedes, one can infer that it is likely that this person's income is above average. Let us define $g(z \mid y, \theta)$ as the density of the missing data conditional on the observed choices in the sample. This conditional density is related to the unconditional density through Bayes' identity:

$$h(z \mid y, \theta) = \frac{P(y \mid z, \theta) f(z \mid \theta)}{P(y \mid \theta)}.$$

Stated succinctly, the density of z conditional on observed choices is proportional to the unconditional density of z times the probability of the observed choices given this z. The denominator is simply the normalizing constant, equal to the integral of the numerator. This concept of a conditional distribution should be familiar to readers from Chapter 11.

Now consider estimation. The LL function is based on the information that the researcher has, which does not include the missing data. The LL function is

$$LL(\theta) = \log P(y \mid \theta) = \log \left(\int P(y \mid z, \theta) f(z \mid \theta) dz \right).$$

In principle, this function can be maximized using the procedures described in Chapter 8. However, as we will see, it is often much easier to maximize LL in a different way.

The procedure is iterative, starting with an initial value of the parameters and updating them in a way to be described. Let the trial value of

[1] We assume in this expression that z is continuous, such that the unconditional probability is an integral. If z is discrete, or a mixture of continuous and discrete variables, then the integration is replaced with a sum over the discrete values, or a combination of integrals and sums.

the parameters in a given iteration be denoted θ^t. Let us define a new function at θ^t that is related to LL but utilizes the conditional density h. This new function is

$$\mathcal{E}(\theta \mid \theta^t) = \int h(z \mid y, \theta^t) \log\Big(P(y \mid z, \theta) f(z \mid \theta) \Big) dz,$$

where the conditional density h is calculated using the current trial value of the parameters, θ^t. This function has a specific meaning. Note that the part on the far right, $P(y \mid z, \theta) f(z \mid \theta)$, is the joint probability of the observed choices and the missing data. The log of this joint probability is the LL of the observed choices *and* the missing data combined. This joint LL is integrated over a density, namely, $h(z \mid y, \theta^t)$. Our function $\mathcal{E}$ is therefore an expectation of the joint LL of the missing data and observed choices. It is a specific expectation, namely, the expectation over the density of the missing data conditional on observed choices. Since the conditional density of z depends on the parameters, this density is calculated using the values θ^t. Stated equivalently, $\mathcal{E}$ is the weighted average of the joint LL, using $h(z \mid y, \theta^t)$ as weights.

The EM procedure consists of repeatedly maximizing $\mathcal{E}$. Starting with some initial value of the parameters, the parameters are updated in each iteration by the following formula:

$$\theta^{t+1} = \text{argmax}_\theta \, \mathcal{E}(\theta \mid \theta^t). \tag{14.1}$$

In each iteration, the current values of the parameters, θ^t, are used to calculate the weights h and then the weighted, joint LL is maximized. The name EM derives from the fact that the procedure utilizes an expectation that is maximized.

It is important to recognize the dual role of the parameters in $\mathcal{E}$. First, the parameters enter the joint log-likelihood of the observed choices and the missing data, $\log P(y \mid z, \theta) f(z \mid \theta)$. Second, the parameters enter the conditional density of the missing data, $h(z \mid y, \theta)$. The function $\mathcal{E}$ is maximized with respect to the former holding the latter constant. That is, $\mathcal{E}$ is maximized over the θ entering $\log P(y \mid z, \theta) f(z \mid \theta)$, holding the θ that enters the weights $h(z \mid y, \theta)$ at their current values θ^t. To denote this dual role, $\mathcal{E}(\theta \mid \theta^t)$ is expressed as a function of θ, its argument over which maximization is performed, given θ^t – the value used in the weights that are held fixed during maximization.

Under very general conditions, the iterations defined by equation (14.1) converge to the maximum of LL. Bolyes (1983) and Wu (1983) provide formal proofs. I provide an intuitive explanation in the next section. However, readers who are interested in seeing examples of the algorithm first can proceed directly to Section 14.3.

14.2.1. Why the EM Algorithm Works

The relation of the EM algorithm to the LL function can be explained in three steps. Each step is a bit opaque, but the three combined provide a startlingly intuitive understanding.

Step 1: Adjust $\mathcal{E}$ to equal LL at θ^t

$\mathcal{E}(\theta \mid \theta^t)$ is not the same as $LL(\theta)$. To aid comparison between them, let us add a constant to $\mathcal{E}(\theta \mid \theta^t)$ that is equal to the difference between the two functions at θ^t:

$$\mathcal{E}^*(\theta \mid \theta^t) = \mathcal{E}(\theta \mid \theta^t) + [LL(\theta^t) - \mathcal{E}(\theta^t \mid \theta^t)].$$

The term in brackets is constant with respect to θ and so maximization of $\mathcal{E}^*$ is the same as maximization of $\mathcal{E}$ itself. Note, however, that by construction, $\mathcal{E}^*(\theta \mid \theta^t) = LL(\theta)$ at $\theta = \theta^t$.

Step 2: Note that the derivative with respect to θ is the same for $\mathcal{E}^*$ and LL evaluated at $\theta = \theta^t$

Consider the derivative of $\mathcal{E}^*(\theta \mid \theta^t)$ with respect to its argument θ:

$$\begin{aligned}\frac{d\mathcal{E}^*(\theta \mid \theta^t)}{d\theta} &= \frac{d\mathcal{E}(\theta \mid \theta^t)}{d\theta} \\ &= \int h(z \mid y, \theta^t) \left(\frac{d \log P(y \mid z, \theta) f(z \mid \theta)}{d\theta} \right) dz \\ &= \int h(z \mid y, \theta^t) \frac{1}{P(y \mid z, \theta) f(z \mid \theta)} \frac{dP(y \mid z, \theta) f(z \mid \theta)}{d\theta} dz.\end{aligned}$$

Now evaluate this derivative at $\theta = \theta^t$:

$$\begin{aligned}&\left. \frac{d\mathcal{E}^*(\theta \mid \theta^t)}{d\theta} \right|_{\theta^t} \\ &= \int h(z \mid y, \theta^t) \frac{1}{P(y \mid z, \theta^t) f(z \mid \theta^t)} \left(\frac{dP(y \mid z, \theta) f(z \mid \theta)}{d\theta} \right)_{\theta^t} dz \\ &= \int \frac{P(y \mid z, \theta^t) f(z \mid \theta^t)}{P(y \mid \theta^t)} \frac{1}{P(y \mid z, \theta^t) f(z \mid \theta^t)} \\ &\quad \times \left(\frac{dP(y \mid z, \theta) f(z \mid \theta)}{d\theta} \right)_{\theta^t} dz \\ &= \int \frac{1}{P(y \mid \theta^t)} \left(\frac{dP(y \mid z, \theta) f(z \mid \theta)}{d\theta} \right)_{\theta^t} dz\end{aligned}$$

$$= \frac{1}{P(y \mid \theta^t)} \int \left(\frac{dP(y \mid z, \theta) f(z \mid \theta)}{d\theta} \right)_{\theta^t} dz$$
$$= \left(\frac{d \log P(y \mid \theta)}{d\theta} \right)_{\theta^t}$$
$$= \left(\frac{dLL(\theta)}{d\theta} \right)_{\theta^t}.$$

At $\theta = \theta^t$, the two functions, $\mathcal{E}^*$ and LL, have the same slope.

Step 3: Note that $\mathcal{E}^* \leq LL$ for all θ
This relation can be shown as follows:

$$\begin{aligned}
& LL(\theta) \\
(14.2) \quad & = \log P(y \mid \theta) \\
& = \log \int P(y \mid z, \theta) f(z \mid \theta) dz \\
& = \log \int \frac{P(y \mid z, \theta) f(z \mid \theta)}{h(z \mid y, \theta^t)} h(z \mid y, \theta^t) dz \\
(14.3) \quad & \geq \int h(z \mid y, \theta^t) \log \frac{P(y \mid z, \theta) f(z \mid \theta)}{h(z \mid y, \theta^t)} dz \\
& = \int h(z \mid y, \theta^t) \log \Big(P(y \mid z, \theta) f(z \mid \theta) \Big) dz \\
& \quad - \int h(z \mid y, \theta^t) \log \Big(h(z \mid y, \theta^t) \Big) dz \\
& = \mathcal{E}(\theta \mid \theta^t) - \int h(z \mid y, \theta^t) \log \Big(h(z \mid y, \theta^t) \Big) dz \\
& = \mathcal{E}(\theta \mid \theta^t) - \int h(z \mid y, \theta^t) \\
& \quad \times \log \left(h(z \mid y, \theta^t) \frac{P(y \mid \theta^t)}{P(y \mid \theta^t)} \right) dz \\
& = \mathcal{E}(\theta \mid \theta^t) + \int h(z \mid y, \theta^t) \log P(y \mid \theta^t) dz \\
& \quad - \int h(z \mid y, \theta^t) \log \Big(h(z \mid y, \theta^t) P(y \mid \theta^t) \Big) dz \\
& = \mathcal{E}(\theta \mid \theta^t) + \log P(y \mid \theta^t) \int h(z \mid y, \theta^t) dz \\
& \quad - \int h(z \mid y, \theta^t) \log \Big(h(z \mid y, \theta^t) P(y \mid \theta^t) \Big) dz
\end{aligned}$$

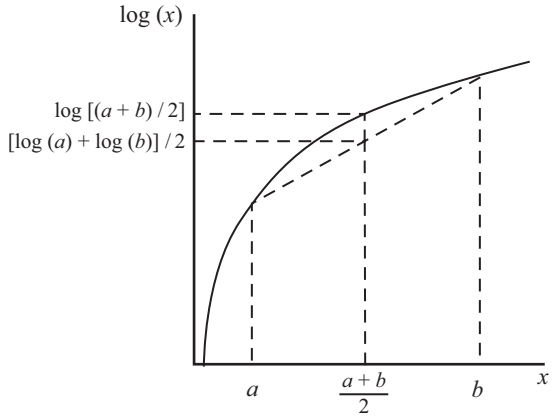

Figure 14.1. Example of Jensen's inequality.

$$
\begin{aligned}
(14.4) \qquad &= \mathcal{E}(\theta \mid \theta^t) + \log P(y \mid \theta^t) - \int h(z \mid y, \theta^t) \\
&\quad \times \log\Big(h(z \mid y, \theta^t) P(y \mid \theta^t)\Big) dz \\
(14.5) \qquad &= \mathcal{E}(\theta \mid \theta^t) + LL(\theta^t) - \int h(z \mid y, \theta^t) \\
&\quad \times \log\Big(P(y \mid z, \theta^t) f(z \mid \theta^t)\Big) dz \\
&= \mathcal{E}(\theta \mid \theta^t) + LL(\theta^t) - \mathcal{E}(\theta^t \mid \theta^t) \\
&= \mathcal{E}^*(\theta \mid \theta^t).
\end{aligned}
$$

The inequality in equation (14.3) is due to Jensen's inequality, which states that $\log(E(x)) > E(\log(x))$. In our case, x is the statistic $\frac{P(y|z,\theta) f(z|\theta)}{h(z|y,\theta^t)}$ and the expectation is over density $h(z \mid y, \theta^t)$. An example of this inequality is shown in Figure 14.1, where the averages are over two values labeled a and b. The average of $\log(a)$ and $\log(b)$ is the midpoint of the dotted line that connects these two points on the log curve. The log evaluated at the average of a and b is $\log((a + b)/2)$, which is above the midpoint of the dotted line. Jensen's inequality is simply a result of the concave shape of the log function.

Equation (14.4) is obtained because the density h integrates to 1. Equation (14.5) is obtained by substituting $h(z \mid y, \theta^t) = P(y \mid z, \theta^t) f(z \mid \theta^t) / P(y \mid \theta^t)$ within the log and then cancelling the $P(y \mid \theta^t)$'s.

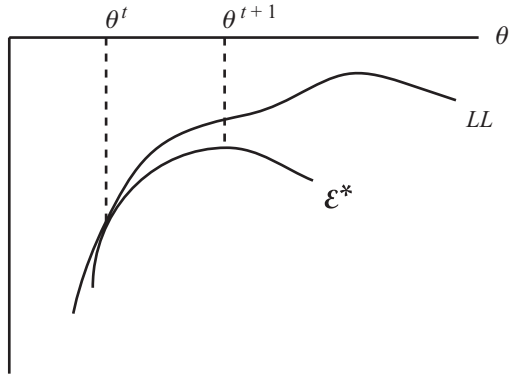

Figure 14.2. Relation of $\mathcal{E}^*$ and LL.

Combine Results to Compare $\mathcal{E}^*$ and LL

Figure 14.2 shows $\mathcal{E}^*(\theta \mid \theta^t)$ and $LL(\theta)$ in the appropriate relation to each other. As we have shown, these two functions are equal and have the same slope at $\theta = \theta^t$. These results imply that the two functions are tangent to each other at $\theta = \theta^t$. We have also shown that $\mathcal{E}^*(\theta \mid \theta^t) \leq LL(\theta)$ for all θ. Consistent with this relation, $\mathcal{E}^*$ is drawn below $LL(\theta)$ in the graph at all points except θ^t where they are the same.

The EM algorithm maximizes $\mathcal{E}^*(\theta \mid \theta^t)$ to find the next trial value of θ. The maximizing value is shown as θ^{t+1}. As the graph indicates, the LL function is necessarily higher at the new parameter value, θ^{t+1}, than at the original value, θ^t. As long as the derivative of the LL function is not already zero at θ^t, maximizing $\mathcal{E}^*(\theta \mid \theta^t)$ raises $LL(\theta)$.[2] Each iteration of the EM algorithm raises the LL function until the algorithm converges at the maximum of the LL function.

14.2.2. Convergence

Convergence of the EM algorithm is usually defined as a sufficiently small change in the parameters (e.g., Levine and Casella, 2001) or in the LL function (e.g., Weeks and Lange, 1989; Aitkin and Aitkin, 1996). These criteria need to be used with care, since the EM algorithm can move slowly near convergence. Ruud (1991) shows that the convergence statistic in Section 8.4 can be used with the gradient and Hessian of $\mathcal{E}$ instead of LL. However, calculating this statistic can be more computationally intensive than the iteration of the EM algorithm itself, and can be infeasible in some cases.

[2] In fact, any increase in $\mathcal{E}^*(\theta \mid \theta^t)$ leads to an increase in $LL(\theta)$.

14.2.3. Standard Errors

There are three ways that standard errors can be calculated. First, once the maximum of $LL(\theta)$ has been found with the EM algorithm, the standard errors can be calculated from LL the same as if the LL function had been maximized directly. The procedures in Section 8.6 are applicable: The asymptotic standard errors can be calculated from the Hessian or from the variance of the observation-specific gradients (i.e., the scores), calculated from $LL(\theta)$ evaluated at $\hat{\theta}$.

A second option arises from the result we obtained in step 2. We showed that $\mathcal{E}$ and LL have the same gradients at $\theta = \theta^t$. At convergence, the value of θ does not change from one iteration to the next, such that $\hat{\theta} = \theta^{t+1} = \theta^t$. Therefore, at $\hat{\theta}$, the derivatives of these two functions are the same. This fact implies that the scores can be calculated from $\mathcal{E}$ rather than LL. If $\mathcal{E}$ takes a more convenient form than LL, as is usually the case when applying an EM algorithm, this alternative calculation can be attractive.

A third option is bootstrap, as also discussed in Section 8.6. Under this option, the EM algorithm is applied numerous times, using a different sample of the observations each time. In many contexts for which EM algorithms are applied, bootstrapped standard errors are more feasible and useful than the asymptotic formulas. The case study in the last section provides an example.

14.3 Examples of EM Algorithms

We describe in this section several types of discrete choice models whose LL functions are difficult to maximize directly but are easy to estimate with EM algorithms. The purpose of the discussion is to provide concrete examples of how EM algorithms are specified as well as to illustrate the value of the approach.

14.3.1. Discrete Mixing Distribution with Fixed Points

One of the issues that arises with mixed logit models (or, indeed, with any mixed model) is the appropriate specification of the mixing distribution. It is customary to use a convenient distribution, such as normal or lognormal. However, it is doubtful that the true distribution of coefficients takes a mathematically convenient form. More flexible distributions are useful, where flexibility means that the specified distribution can take a wider variety of shapes, depending on the values of its parameters.

Usually, greater flexibility is attained by including more parameters. In nonparametric estimation, a family of distributions is specified that has the property that the distribution becomes more flexible as the number of parameters rises. By allowing the number of parameters to rise with sample size, the nonparametric estimator is consistent for any true distribution. The term "nonparametric" is a bit of a misnomer in this context; "superparametric" is perhaps more appropriate, since the number of parameters is usually larger than that under standard specifications and rises with sample size to gain ever-greater flexibility.

The large number of parameters in nonparametric estimation makes direct maximization of the LL function difficult. In many cases, however, an EM algorithm can be developed that facilitates estimation considerably. The current section presents one such case.

Consider a mixed logit with an unknown distribution of coefficients. Any distribution can be approximated arbitrarily closely by a discrete distribution with a sufficiently large number of points. We can use this fact to develop a nonparametric estimator of the mixing distribution, using an EM algorithm for estimation.

Let the density of coefficients be represented by C points, with β_c being the cth point. The location of these points is assumed (for the current procedure) to be fixed, and the mass at each point (i.e., the share of the population at each point) is the parameter to be estimated. One way to select the points is to specify a maximum and minimum for each coefficient and create a grid of evenly spaced points between the maxima and minima. For example, suppose there are five coefficients and the range between the minimum and maximum of each coefficient is represented by 10 evenly spaced points. The 10 points in each dimension create a grid of $10^5 = 100{,}000$ points in the five-dimensional space. The parameters of the model are the share of the population at each of the 100,000 points. As we will see, estimation of such a large number of parameters is quite feasible with an EM algorithm. By increasing the number of points, the grid becomes ever finer, such that the estimation of shares at the points approximates any underlying distribution.

The utility that agent n obtains from alternative j is

$$U_{nj} = \beta_n x_{nj} + \varepsilon_{nj},$$

where ε is i.i.d. extreme value. The random coefficients have the discrete distribution described previously, with s_c being the share of the population at point β_c. The distribution is expressed functionally as

$$f(\beta_n) = \begin{cases} s_1 & \text{if } \beta_n = \beta_1 \\ s_2 & \text{if } \beta_n = \beta_2 \\ \vdots & \\ s_C & \text{if } \beta_n = \beta_C \\ 0 & \text{otherwise,} \end{cases}$$

where the shares sum to 1: $\sum_c s_c = 1$. For convenience, denote the shares collectively as vector $s = \langle s_1, \ldots, s_C \rangle$.[3]

Conditional on $\beta_n = \beta_c$ for some c, the choice model is a standard logit:

$$L_{ni}(\beta_c) = \frac{e^{\beta_c x_{ni}}}{\sum_j e^{\beta_c x_{nj}}}.$$

Since β_n is not known for each person, the choice probability is a mixed logit, mixed over the discrete distribution of β_n:

$$P_{ni}(s) = \sum_c s_c L_{ni}(\beta_c).$$

The LL function is $LL(s) = \sum_n \log P_{ni_n}(s)$, where i_n is the chosen alternative of agent n.

The LL can be maximized directly to estimate the shares s. With a large number of classes, as would usually be needed to flexibly represent the true distribution, this direct maximization can be difficult. However, an EM algorithm can be utilized for this model that is amazingly simple, even with hundreds of thousands of points.

The "missing data" in this model are the coefficients of each agent. The distribution f gives the share of the population with each coefficient value. However, as discussed in Chapter 11, a person's choice reveals information about their coefficients. Conditional on person n choosing alternative i_n, the probability that the person has coefficients β_c is given by Bayes' identity:

$$h(\beta_c \mid i_n, s) = \frac{s_c L_{ni_n}(\beta_c)}{P_{ni_n}(s)}.$$

[3] This specification can be considered a type of latent class model, where there are C classes, the coefficients of people in class c are β_c, and s_c is the share of the population in class c. However, the term "latent class model" usually refers to a model in which the location of the points are parameters as well as the shares. We consider this more traditional form in our next example.

This conditional distribution is used in the EM algorithm. In particular, the expectation for the EM algorithm is

$$\mathcal{E}(s \mid s^t) = \sum_n \sum_c h(\beta_c \mid i_n, s^t) \log\Big(s_c L_{ni_n}(\beta_c)\Big).$$

Since $\log\Big(s_c\, L_{ni_n}(\beta_c)\Big) = \log(s_c) + \log L_{ni_n}(\beta_c)$, this expectation can be rewritten as two parts:

$$\begin{aligned}\mathcal{E}(s \mid s^t) = &\sum_n \sum_c h(\beta_c \mid i_n, s^t) \log(s_c) \\ &+ \sum_n \sum_c h(\beta_c \mid i_n, s^t) \log L_{ni_n}(\beta_c).\end{aligned}$$

This expectation is to be maximized with respect to the parameters s. Note, however, that the second part on the right does not depend on s: it depends only on the coefficients β_c that are fixed points in this nonparametric procedure. Maximization of the preceding formula is therefore equivalent to maximization of just the first part:

$$\mathcal{E}(s \mid s^t) = \sum_n \sum_c h(\beta_c \mid i_n, s^t) \log(s_c).$$

This function is very easy to maximize. In particular, the maximizing value of s_c, accounting for the constraint that the shares sum to 1, is

$$s_c^{t+1} = \frac{\sum_n h(\beta_c \mid i_n, s^t)}{\sum_n \sum_{c'} h(\beta_{c'} \mid i_n, s^t)}.$$

Using the nomenclature from the general description of EM algorithms, $h(\beta_c \mid i_n, s^t)$ are weights, calculated at the current value of the shares s^t. The updated share for class c is the sum of weights at point c as a share of the sum of weights at all points.

This EM algorithm is implemented in the following steps:

1. Define the points β_c for $c = 1, \ldots, C$.
2. Calculate the logit formula for each person at each point: $L_{ni}(\beta_c)\ \forall n, c$.
3. Specify initial values of the share at each point, labeled collectively as s^0. It is convenient for the initial shares to be $s_c = 1/C\ \forall c$.
4. For each person and each point, calculate the probability of the person having those coefficients conditional on the person's choice, using the initial shares s^0 as the unconditional probabilities: $h(\beta_c \mid i_n, s^0) = s_c^0 L_{ni}(\beta_c)/P_{ni}(s^0)$. Note that the

denominator is the sum over points of the numerator. For convenience, label this calculated value h^0_{nc}.

5. Update the population share at point c as $s^1_c = \frac{\sum_n h^0_{nc}}{\sum_n \sum_{c'} h^0_{nc'}}$.
6. Repeat steps 4 and 5 using the updated shares s in lieu of the original starting values. Continue repeating until convergence.

This procedure does not require calculation of any gradients or inversion of any Hessian, as the procedures in Chapter 8 utilize for direct maximization of LL. Moreover, the logit probabilities are calculated only once (in step 2), rather than in each iteration. The iterations consist of recalibrating the shares at each point, which is simple arithmetic. Because so little calculation is needed for each point, the procedure can be implemented with a very large number of points. For example, the application in Train (2008a) included more than 200,000 points, and yet estimation took only about 30 minutes. In contrast, it is doubtful that direct maximization by the methods in Chapter 8 would have even been feasible, since they would entail inverting a 200,000 × 200,000 Hessian.

14.3.2. Discrete Mixing Distribution with Points as Parameters

We can modify the previous model by treating the coefficients, β_c for each c, as parameters to be estimated rather than fixed points. The parameters of the model are then the location and share of the population at each point. Label these parameters collectively as $\theta = \langle s_c, \beta_c, c = 1, \ldots, C\rangle$. This specification is often called a *latent class model*: the population consists of C distinct classes with all people within a class having the same coefficients but coefficients being different for different classes. The parameters of the model are the share of the population in each class and the coefficients for each class.

The missing data are the class membership of each person. The expectation for the EM algorithm is the same as for the previous specification except that now the β_c's are treated as parameters:

$$\mathcal{E}(\theta \mid \theta^t) = \sum_n \sum_c h(\beta_c \mid i_n, s^t) \log\left(s_c L_{ni_n}(\beta_c)\right).$$

Note that each set of parameters enters only one term inside the log: the vector of shares s does not enter any of the $L_{ni_n}(\beta_c)$'s, and each β_c enters only $L_{ni_n}(\beta_c)$ for class c. Maximization of this function is therefore equivalent to separate maximization of each of the following functions:

$$\mathcal{E}(s \mid \theta^t) = \sum_n \sum_c h(\beta_c \mid i_n, s^t) \log(s_c) \tag{14.6}$$

and for each c:

$$\mathcal{E}(\beta_c \mid \theta^t) = \sum_n h(\beta_c \mid i_n, s^t) \log L_{ni_n}(\beta_c). \tag{14.7}$$

The maximum of (14.6) is attained, as before, at

$$s_c^{t+1} = \frac{\sum_n h(\beta_c \mid i_n, s^t)}{\sum_n \sum_{c'} h(\beta_{c'} \mid i_n, s^t)}.$$

For updating the coefficients β_c, note that (14.7) is the LL function for a standard logit model, with each observation weighted by $h(\beta_c \mid i_n, s^t)$. The updated values of β_c are obtained by estimating a standard logit model with each person providing an observation that is weighted appropriately. The same logit estimation is performed for each class c, but with different weights for each class.

The EM algorithm is implemented in these steps:

1. Specify initial values of the share and coefficients in each class, labeled $\beta_c^0 \; \forall c$ and s^0. It is convenient for the initial shares to be $1/C$. I have found that starting values for the coefficients can easily be obtained by partitioning the sample into C groups and running a logit on each group.[4]
2. For each person and each class, calculate the probability of being in that class conditional on the person's choice, using the initial shares s^0 as the unconditional probabilities: $h(\beta_c^0 \mid i_n, s^0) = s_c^0 L_{ni_n}(\beta_c^0)/P_{ni_n}(s^0)$. Note that the denominator is the sum over classes of the numerator. For convenience, label this calculated value h_{nc}^0.
3. Update the share in class c as $s_c^1 = \frac{\sum_n h_{nc}^0}{\sum_n \sum_{c'} h_{nc'}^0}$.
4. Update the coefficients for each class c by estimating a logit model with person n weighted by h_{nc}^0. A total of C logit models are estimated, using the same observations but different weights in each.
5. Repeat steps 2–4 using the updated shares s and coefficients $\beta_c \; \forall c$ in lieu of the original starting values. Continue repeating until convergence.

[4] Note that these groups do not represent a partitioning of the sample into classes. The classes are latent and so such partitioning is not possible. Rather, the goal is to obtain C sets of starting values for the coefficients of the C classes. These starting values must not be the same for all classes since, if they were the same, the algorithm would perform the same calculations for each class and return for all classes the same shares and updated estimates in each iteration. An easy way to obtain C different sets of starting values is to divide the sample into C groups and estimate a logit on each group.

An advantage of this approach is that the researcher can implement non-parametric estimation, with the class shares and coefficients in each class treated as parameters, using any statistical package that includes a logit estimation routine. For a given number of classes, this procedure takes considerably longer than the previous one, since C logit models must be estimated in each iteration. However, far fewer classes are probably needed with this approach than the previous one to adequately represent the true distribution, since this procedure estimates the "best" location of the points (i.e, the coefficients), while the previous one takes the points as fixed.

Bhat (1997a) developed an EM algorithm for a model that is similar to this one, except that the class shares s_c are not parameters in themselves but rather are specified to depend on the demographics of the person. The EM algorithm that he used replaces our step 3 with a logit model of class membership, with conditional probability of class membership serving as the dependent variable. His application demonstrates the overriding point of this chapter: that EM algorithms can readily be developed for many forms of complex choice models.

14.3.3. Normal Mixing Distribution with Full Covariance

We pointed out in Chapter 12 that a mixed logit with full covariance among coefficients can be difficult to estimate by standard maximization of the LL function, due both to the large number of covariance parameters and to the fact that the LL is highly non-quadratic. Train (2008b) developed an EM algorithm that is very simple and fast for mixed logits with full covariance. The algorithm takes the following very simple form:

1. Specify initial values for the mean and covariance of the coefficients in the population.
2. For each person, take draws from the population distribution using this initial mean and covariance.
3. Weight each person's draws by that person's conditional density of the draws.
4. Calculate the mean and covariance of the weighted draws for all people. These become the updated mean and covariance of the coefficients in the population.
5. Repeat steps 2–4 using the updated mean and covariance, and continue repeating until there is no further change (to a tolerance) in the mean and covariance.

The converged values are the estimates of the population mean and covariance. No gradients are required. All that is needed for estimation of a model with fully correlated coefficients is to repeatedly take draws using the previously calculated mean and covariance, weight the draws appropriately, and calculate the mean and covariance of the weighted draws.

The procedure is readily applicable when the coefficients are normally distributed or when they are transformations of jointly normal terms. Following the notation in Train and Sonnier (2005), the utility that agent n obtains from alternative j is

$$U_{nj} = \alpha_n x_{nj} + \varepsilon_{nj},$$

where the random coefficients are transformations of normally distributed terms: $\alpha_n = T(\beta_n)$, with β_n distributed normally with mean b and covariance W. The transformation allows considerable flexibility in the choice of distribution. For example, a lognormal distribution is obtained by specifying transformation $\alpha_n = \exp(\beta_n)$. An S_b distribution, which has an upper and lower bound, is obtained by specifying $\alpha_n = \exp(\beta_n)/(1 + \exp(\beta_n))$. Of course, if the coefficient is itself normal then $\alpha_n = \beta_n$. The normal density is denoted $\phi(\beta_n \mid b, W)$.

Conditional on β_n, the choice probabilities are logit:

$$L_{ni}(\beta_n) = \frac{e^{T(\beta_n)x_{ni}}}{\sum_j e^{T(\beta_n)x_{nj}}}.$$

Since β_n is not known, the choice probability is a mixed logit, mixed over the distribution of β_n:

$$P_{ni}(b, W) = \int L_{ni}(\beta)\phi(\beta \mid b, W)d\beta.$$

The LL function is $LL(b, W) = \sum_n \log P_{ni_n}(b, W)$, where i_n is the chosen alternative of agent n.

As discussed in Section 12.7, classical estimation of this model by standard maximization of LL is difficult, and this difficulty is one of the reasons for using Bayesian procedures. However, an EM algorithm can be applied that is considerably easier than standard maximization and makes classical estimation about as convenient as Bayesian for this model.

The missing data for the EM algorithm are the β_n of each person. The density $\phi(\beta \mid b, W)$ is the distribution of β in the population. For the EM algorithm, we use the conditional distribution for each person. By Bayes' identity, the density of β conditional on alternative i being

chosen by person n is $h(\beta \mid i, b, W) = L_{ni}(\beta)\phi(\beta \mid b, W)/P_{ni}(b, W)$. The expectation for the EM algorithm is

$$\mathcal{E}(b, W \mid b^t, W^t) = \sum_n \int h(\beta \mid i_n, b^t, W^t) \times \log\Big(L_{ni_n}(\beta)\phi(\beta \mid b, W)\Big)d\beta.$$

Note that $L_{ni_n}(\beta)$ does not depend on the parameters b and W. Maximization of this expectation with respect to the parameters is therefore equivalent to maximization of

$$\mathcal{E}(b, W \mid b^t, W^t) = \sum_n \int h(\beta \mid i_n, b^t, W^t)\log\big(\phi(\beta \mid b, W)\big)d\beta.$$

The integral inside this expectation does not have a closed form. However we can approximate the integral through simulation. Substituting the definition of $h(\cdot)$ and rearranging, we have

$$\mathcal{E}(b, W \mid b^t, W^t) = \sum_n \int \frac{L_{ni_n}(\beta)}{P_{ni_n}(b^t, W^t)}\log\big(\phi(\beta \mid b, W)\big) \times \phi(\beta \mid b^t, W^t)d\beta.$$

The expectation over ϕ is simulated by taking R draws from $\phi(\beta \mid b^t, W^t)$ for each person, labeled β_{nr} for the rth draw for person n. The simulated expectation is

$$\tilde{\mathcal{E}}(b, W \mid b^t, W^t) = \sum_n \sum_r w_{nr}^t \log\big(\phi(\beta_{nr} \mid b, W)\big)/R,$$

where the weights are

$$w_{nr}^t = \frac{L_{ni_n}(\beta_{nr})}{\frac{1}{R}\sum_{r'} L_{ni_n}(\beta_{nr'})}.$$

This simulated expectation takes a familiar form: it is the LL function for a sample of draws from a normal distribution, with each draw weighted by w_{nr}^t.[5] The maximum likelihood estimator of the mean and covariance of a normal distribution, given a weighted sample of draws from that distribution, is simply the weighted mean and covariance of the sampled draws. The updated mean is

$$b^{t+1} = \frac{1}{NR}\sum_n \sum_r w_{nr}^t \beta_{nr}$$

[5] The division by R can be ignored since it does not affect maximization, in the same way that division by sample size N is omitted from $\mathcal{E}$.

and the updated covariance matrix is

$$W^{t+1} = \frac{1}{NR}\sum_n \sum_r w_{nr}^t (\beta_{nr} - b^{t+1})(\beta_{nr} - b^{t+1})'.$$

Note that W^{t+1} is necessarily positive definite, as required for a covariance matrix, since it is constructed as the covariance of the draws.

The EM algorithm is implemented as follows:

1. Specify initial values for the mean and covariance, labeled b^0 and W^0.
2. Create R draws for each of the N people in the sample as $\beta_{nr}^0 = b^0 + chol(W^0)\eta_{nr}$, where $chol(W^0)$ is the lower-triangular Choleski factor of W^0 and η_{nr} is a conforming vector of i.i.d. standard normal draws.
3. For each draw for each person, calculate the logit probability of the person's observed choice: $L_{ni_n}(\beta_{nr}^0)$.
4. For each draw for each person, calculate the weight

$$w_{nr}^0 = \frac{L_{ni_n}(\beta_{nr}^0)}{\sum_{r'} L_{ni_n}(\beta_{nr'}^0)/R}.$$

5. Calculate the weighted mean and covariance of the $N * R$ draws β_{nr}^0, $r = 1, \ldots, R$, $n = 1, \ldots, N$, using weight w_{nr}^0. The weighted mean and covariance are the updated parameters b^1 and W^1.
6. Repeat steps 2–5 using the updated mean b and variance W in lieu of the original starting values. Continue repeating until convergence.

This procedure can be implemented without evoking any estimation software, simply by taking draws, calculating logit formulas to construct weights, and then calculating the weighted mean and covariance of the draws. A researcher can estimate a mixed logit with full covariance and with coefficients that are possibly transformations of normals with just these simple steps.

Train (2008a) shows that this procedure can be generalized to a finite mixture of normals, where β is drawn from any of C normals with different means and covariances. The probability of drawing β from each normal (i.e., the share of the population whose coefficients are described by each normal distribution) is a parameter along with the means and covariances. Any distribution can be approximated by a finite mixture of normals, with a sufficient number of underlying normals. By allowing the number of normals to rise with sample size, the

approach becomes a form of nonparametric estimation of the true mixing distribution. The EM algorithm for this type of nonparametrics combines the concepts of the current section on normal distributions with full covariance and the concepts of the immediately previous section on discrete distributions.

One last note is useful. At convergence, the derivatives of $\tilde{\mathcal{E}}(b, W \mid \hat{b}, \hat{W})$ provide scores that can be used to estimate the asymptotic standard errors of the estimates. In particular, the gradient with respect to b and W is

$$\frac{\tilde{\mathcal{E}}(b, W \mid \hat{b}, \hat{W})}{db} = \sum_n \left[\frac{1}{R} \sum_r -w_{nr} W^{-1}(\beta_{nr} - b) \right]$$

and

$$\frac{\tilde{\mathcal{E}}(b, W \mid \hat{b}, \hat{W})}{dW} = \sum_n \left[\frac{1}{R} \sum_r w_{nr} \left(-\frac{1}{2} W^{-1} + \frac{1}{2} W^{-1}(\beta_{nr} - b)(\beta_{nr} - b)' W^{-1} \right) \right],$$

where the weights w_{nr} are calculated at the estimated values $\hat{b}$ and $\hat{W}$. The terms in brackets are the scores for each person, which we can collect into a vector labeled s_n. The variance of the scores is $V = \sum_n s_n s_n' / N$. The asymptotic covariance of the estimator is then calculated as V^{-1}/N, as discussed in Section 8.6.

14.4 Case Study: Demand for Hydrogen Cars

Train (2008a) examined buyers' preferences for hydrogen-powered vehicles, using several of the EM algorithms that we have described. We will describe one of his estimated models as an illustration of the procedure. A survey of new car buyers in Southern California was conducted to assess the importance that these buyers place on several issues that are relevant for hydrogen vehicles, such as the availability of refueling stations. Each respondent was presented with a series of 10 stated-preference experiments. In each experiment the respondent was offered a choice among three alternatives: the conventional-fuel vehicle (CV) that the respondent had recently purchased and two alternative-fuel vehicles (AVs) with specified attributes. The respondent was asked to evaluate the three options, stating which they considered best and which worst. The attributes represented relevant features of hydrogen vehicles, but the respondents were not told that the alternative fuel was hydrogen so as to avoid any preconceptions that respondents might have

developed with respect to hydrogen vehicles. The attributes included in the experiments are as follows:

- Fuel cost (FC), expressed as percent difference from the CV. In estimation, the attribute was scaled as a share, such that fuel costs of 50 percent less than the conventional-fuel vehicle enters as -0.5 and 50 percent more enters as 0.5.
- Purchase price (PP), expressed as percent difference from the CV, scaled analogously to fuel cost when entering the model.
- Driving radius (DR): the farthest distance from home that one is able to travel and then return, starting on a full tank of fuel. As defined, driving radius is one-half the vehicle's range. In the estimated models, DR was scaled in hundreds of miles.
- Convenient medium-distance destinations (CMDD): the percent of destinations within the driving radius that "require no advanced planning because you can refuel along the way or at your destination" as opposed to destinations that "require refueling (or at least estimating whether you have enough fuel) before you leave to be sure you can make the round-trip." This attribute reflects the distribution of potential destinations and refueling stations within the driving radius, recognizing that the tank will not always be full when starting. In the estimated models, it was entered as a share, such that, for example, 50 percent enters as 0.50.
- Possible long-distance destinations (PLDD): the percent of destinations beyond the driving radius that are possible to reach because refueling is possible, as opposed to destinations that cannot be reached due to limited station coverage. This attribute reflects the extent of refueling stations outside the driving radius and their proximity to potential driving destinations. It entered the models scaled analogously to CMDD.
- Extra time to local stations (ETLS): additional one-way travel time, beyond the time typically required to find a conventional fuel station, that is required to get to an alternative fuel station in the local area. ETLS was defined as having values of 0, 3, and 10 minutes in the experiments; however, in preliminary analysis, it was found that respondents considered 3 minutes to be no inconvenience (i.e., equivalent to 0 minutes). In the estimated models, therefore, a dummy variable for ETLS being 10 or not was entered, rather than ETLS itself.

In the experiments, the CV that the respondent had purchased was described as having a driving radius of 200 miles, CMDD and PLDD equal to 100 percent, and, by definition, ETLS, FC, and PP of 0.

Table 14.1. *Mixed logit models with discrete distribution of coefficients and different numbers of classes*

Classes	Log-Likelihood	Parmeters	AIC	BIC
1	−7,884.6	7	15,783.2	15,812.8
5	−6,411.5	39	12,901.0	13,066.0
6	−6,335.3	47	12,764.6	12,963.4
7	−6,294.4	55	12,698.8	12,931.5
8	−6,253.9	63	12,633.8	12,900.3
9	−6,230.4	71	12,602.8	12,903.2
10	−6,211.4	79	12,580.8	12,915.0
15	−6,124.5	119	12,487.0	12,990.4
20	−6,045.1	159	12,408.2	13,080.8
25	−5,990.7	199	12,379.4	13,221.3
30	−5,953.4	239	12,384.8	13,395.9

As stated previously, the respondent was asked to identify the best and worst of the three alternatives, thereby providing a ranking of the three. Conditional on the respondent's coefficients, the ranking probabilities were specified with the "exploded logit" formula (as described in Section 7.3.1). By this formulation, the probability of the ranking is the logit probability of the first choice from the three alternatives in the experiment, times the logit probability for the second choice from the two remaining alternatives. This probability is mixed over the distribution of coefficients, whose parameters were estimated.

All three methods that we describe previously were applied. We concentrate on the method in Section 14.3.2, since it provides a succinct illustration of the power of the EM algorithm. For this method, there are C classes of buyers, and the coefficients β_n and share s_c of the population in each class are treated as parameters.

Train (2008a) estimated the model with several different numbers of classes, ranging from 1 class (which is a standard logit) up to 30 classes. Table 14.1 gives the LL value for these models. Increasing the number of classes improves LL considerably, from −7,884.6 with 1 class to −5,953.4 with 30 classes. Of course, a larger number of classes entail more parameters, and the question arises of whether the improved fit is "worth" the extra parameters. In situations such as this, it is customary to evaluate the models by the Akaike information criterion (AIC) or Bayesian information criterion (BIC).[6] The values of these statistics are

[6] See, e.g., Mittelhammer et. al. (2000, section 18.5) for a discussion of information criteria. The AIC (Akaike, 1974) is $-2LL + 2K$, where LL is the value of the log-likelihood and K is the number of parameters. The BIC, also called Schwarz (1978) criterion, is $-2LL + \log(N)K$, where N is sample size.

Table 14.2. *Model with eight classes*

Class	1	2	3	4
Shares	0.107	0.179	0.115	0.0699
Coefficients				
FC	−3.546	−2.576	−1.893	−1.665
PP	−2.389	−5.318	−12.13	0.480
DR	0.718	0.952	0.199	0.472
CMDD	0.662	1.156	0.327	1.332
PLPP	0.952	2.869	0.910	3.136
ETLS = 10 dummy	−1.469	−0.206	−0.113	−0.278
CV dummy	−1.136	−0.553	−0.693	−2.961
Class	5	6	7	8
Shares	0.117	0.077	0.083	0.252
Coefficients				
FC	−1.547	−0.560	−0.309	−0.889
PP	−2.741	−1.237	−1.397	−2.385
DR	0.878	0.853	0.637	0.369
CMDD	0.514	3.400	−0.022	0.611
PLPP	0.409	3.473	0.104	1.244
ETLS = 10 dummy	0.086	−0.379	−0.298	−0.265
CV dummy	−3.916	−2.181	−0.007	2.656

also given in Table 14.1. The AIC is lowest (best) with 25 classes, and the BIC, which penalizes extra parameters more heavily than the AIC, is lowest with 8 classes.

For the purposes of evaluating the EM algorithm, it is useful to know that these models were estimated with run times of about 1.5 minutes per classes, from initial values to convergence. This means that the model with 30 classes, which has 239 parameters,[7] was estimated in only 45 minutes.

Table 14.2 presents the estimates for the model with 8 classes, which is best by the BIC. The model with 25 classes, which is best by the AIC, provides even greater detail but is not given for the sake of brevity. As shown in Table 14.2, the largest of the 8 classes is the last one with 25 percent. This class has a large, positive coefficient for CV, unlike all the other classes. This class apparently consists of people who prefer their CV over AVs even when the AV has the same attributes, perhaps because of the uncertainty associated with new technologies. Other distinguishing features of classes are evident. For example, class 3 cares

[7] Seven coefficients and 1 share for each of 30 classes, with one class share determined by constraint that the shares sum to 1.

Table 14.3. *Summary statistics for coefficients*

	Means		Standard deviation	
	Est.	SE	Est.	SE
Coefficients				
FC	−1.648	0.141	0.966	0.200
PP	−3.698	0.487	3.388	0.568
DR	0.617	0.078	0.270	0.092
CMDD	0.882	0.140	0.811	0.126
PLPP	1.575	0.240	1.098	0.178
ETLS = 10 dummy	−0.338	0.102	0.411	0.089
CV dummy	−0.463	1.181	2.142	0.216

Est., estimate; SE, standard error.

far more about PP than the other classes, while class 1 places more importance on FC than the other classes.

Table 14.3 gives the mean and standard deviation of the coefficients over the 8 classes. As Train (2008a) points out, these means and standard deviations are similar to those obtained with a more standard mixed logit with normally distributed coefficients (shown in his paper but not repeated here). This result indicates that the use of numerous classes in this application, which the EM algorithm makes possible, provides greater detail in explaining differences in preferences while maintaining very similar summary statistics.

It would be difficult to calculate standard errors from asymptotic formulas for this model (i.e., as the inverse of the estimated Hessian), since the number of parameters is so large. Also, we are interested in summary statistics, such as the mean and standard deviation of the coefficients over all classes, given in Table 14.4. Deriving standard errors for

Table 14.4. *Standard errors for class 1*

	Est.	SE
Share	0.107	0.0566
Coefficients		
FC	−3.546	2.473
PP	−2.389	6.974
DR	0.718	0.404
CMDD	0.662	1.713
PLPP	0.952	1.701
ETLS = 10 dummy	−1.469	0.956
CV dummy	−1.136	3.294

Est., estimate; SE, standard error.

these summary statistics from asymptotic formulas for the covariance of the parameters themselves would be computationally difficult. Instead, standard errors can readily be calculated for this model by bootstrap. Given the speed of the EM algorithm in this application, bootstrapping is quite feasible. Also, bootstrapping automatically provides standard errors for our summary statistics (by calculating the summary statistics for each bootstrap estimate and taking their standard deviations).

The standard errors for the summary statistics are given in Table 14.3, based on 20 bootstrapped samples. Standard errors are not given in Table 14.2 for each class's parameters. Instead, Table 14.4 gives standard errors for class 1, which is illustrative of all the classes. As the table shows, the standard errors for the class 1 parameters are large. These large standard errors are expected and arise from the fact that the labeling of classes in this model is arbitrary. Suppose, as an extreme but illustrative example, that two different bootstrap samples give the same estimates for two classes but with their order changed (i.e., the estimates for class 1 becoming the estimates for class 2, and vice versa). In this case, the bootstrapped standard errors for the parameters for both classes rise even though the model for these two classes together is exactly the same. Summary statistics avoid this issue. All but one of the means are statistically significant, with the CV dummy obtaining the only insignificant mean. All of the standard deviations are significantly different from zero.

Train (2008a) also estimated two other models on these data using EM algorithms: (1) a model with a discrete distribution of coefficients where the points are fixed and the share at each point is estimated, using the procedure in Section 14.3.1, and (2) a model with a discrete mixture of two normal distributions with full covariance, using a generalization of the procedure in Section 14.3.3. The flexibility of EM algorithms to accommodate a wide variety of complex models is the reason why they are worth learning. They enhance the researcher's ability to build specially tailored models that closely fit the reality of the situation and the goals of the research – which has been the overriding objective of this book.