CHAPTER 6

Computation: The NPMLE

We now consider the computational issues involved in calculating the NPMLE. We will focus on the simplified version of the problem in which there are no auxiliary parameters. The standard strategy to incorporate auxiliary parameters is to alternate between an algorithm for the latent parameters and one for the auxiliary parameters; this section describes only the latent parameter phase of that operation.

We will start with an overview of the algorithmic strategies available, deferring the details to the works of others.

After the overview, we wish to address an important issue that has seen little attention. In most problems, one cannot hope to compute the NPMLE exactly because there is no finite time algorithm that will attain the solution. Thus one must devise strategies that ensure that the computations have gone far enough to give desired statistical accuracy, but have not gone needlessly far. We will offer one strategy for this.

At this point, there is a limited supply of software available for the nonparametric analysis. See Böhning, Schlattman and Lindsay (1992) for a description of C.A.MAN, Ezzet and Davies (1988) for a description of MIXTURE and DerSimonian (1986, 1990) for a published algorithm.

6.1. The convergence issue. We recall that the test for whether a *candidate* latent distribution Q is the nonparametric maximum likelihood estimator $\hat{Q}$ is to check whether the gradient inequality holds:

$$D_Q(\phi) \leq 0 \qquad \forall\, \phi \in \Omega.$$

Unfortunately, in a typical problem one has an iterative algorithm such that one cannot in a finite amount of time attain this inequality. There are two issues here.

First, there are often *infinitely* many inequalities to check, corresponding to all ϕ in Ω. We will later consider the implications of a simple solution to this problem where we assume that there is a chosen finite subset, say Ω_s, of s grid points ϕ_j, where the gradient will be checked. One of our points of interest becomes the appropriate choice for the elements of such a grid.

A more sophisticated approach, used by Lesperance and Kalbfleisch (1992), involves taking a basic grid and then doing further searches for gradient violations in the neighborhood of each grid point. An appropriate analysis of such a method requires rather more sophistication than our approach here. The key to a proper basic grid would be that there could not be regions between grid points that were left unsearched. One of the key points of our discussion here is that there are many situations in which we do not need this extra sophistication in programming. We can simply choose a grid to work on and iterate to a solution.

We note that, provided that one restricts the support points of Q to be also from the chosen finite grid, then the distribution Q that solves

$$D_Q(\phi) \leq 0 \qquad \forall\, \phi \in \Omega_s$$

is the nonparametric MLE among all latent distributions on the grid.

Unfortunately, even with such a restricted grid, one must still use an iterative algorithm that does not converge in a finite number of steps. We note in passing that when we restrict the maximization to a finite grid, we are exactly in the setting of known component densities, Section 3.2.1, where there is no known finite step algorithm.

One possible strategy at setting a stopping rule would therefore be to set a small positive tolerance on the values of the gradient function, such as γ, and set a stopping rule for the algorithmic process of the type

$$D_Q(\phi) \leq \gamma \qquad \forall\, \phi \in \Omega_s.$$

We return to this rule later, after reviewing some of the main algorithms that have been employed for this problem.

6.2. Using the EM. The simplest programming approach to finding the NPMLE is to follow the lead of Laird (1978), who suggested the use of the EM algorithm with a large number of support points. For example, one could use one support point for each data point, with locations suggested by the unicomponent maximum likelihood estimators. Provided one has the patience for a very slow algorithm and has a secondary strategy for monitoring the gradient function to ensure adequate convergence, this is a reasonable strategy. We note that even if there is not an explicit solution to the EM equations for the component parameters, one can always simply fix a grid of ϕ values and use the EM to solve for the unknown weights because this algorithm is always simply calculated.

See DerSimonian (1986, 1990) for details of an algorithm.

6.3. Gradient-based algorithms. For completeness, we here offer a description of some basic algorithmic issues and constructions. As more complete sources, we recommend Böhning, Schlattman and Lindsay (1992) and Böhning (1995).

6.3.1. *Design algorithms.* If we view the problem in the wide context as a convex optimization problem on a set defined by its convex hull representation, then the mixture problem is, of course, identical in structure to other problems. Of particular relevance to statisticians is the problem of the *optimal design of experiments*, where the *objective function* is one of several scalar measures of the overall informativeness of an experiment based on the resulting information matrix for the parameters. The *feasible region* is the space of information matrices that are allowed under the design constraints. The space of feasible design matrices can be shown to be the convex hull of a set of basic matrices determined by the selection of a single design point. See Silvey (1980) for an overview of this problem.

Of consequence here is that many of the algorithms used in mixtures can be used in design problems as well, including EM type algorithms, and of course, the converse is true. Thus a detailed study of the literature is necessarily quite extensive, and we will here only point to some of the mixture sources.

6.3.2. *Keeping track of the support points.* One technical issue that separates various algorithmic approaches concerns the kind of information that is stored and used at each iteration. The reason for this is that it is technically not necessary to keep track of the current latent distribution estimator Q_c at each step. If one instead merely updates the current fitted likelihood vector $\mathbf{L}_c$, uses this to construct the gradient function and therefrom an update of $\mathbf{L}_c$, say $\mathbf{L}_{c+1}$, then one may have everything one needs. Since there are a number of algorithms of this type, we need to say why.

It is because $\hat{Q}$ itself can in theory be reconstructed after convergence from $\hat{\mathbf{L}}$. Its support points ξ_j can be found from the final gradient function because they are in the set of points where the gradient reaches a local maximum of zero. The weights π_j, at least in theory, can then be determined after convergence by the relationship

$$\sum \pi_j \mathbf{L}(\xi_j) = \hat{\mathbf{L}}.$$

Such an algorithm might be contrasted with an EM algorithm that uses many support points—an approach recommended by some—because we must save the current parameter values for it so that they can be updated at the next step.

6.3.3. *Vertex direction and exchange methods.* In the following algorithmic methods, we will let Q_c be the current estimate of the latent distribution and we describe the construction of the next step.

The simplest algorithm for finding the NPMLE springs directly from the gradient function itself. Sometimes called the *vertex direction method*, it simply consists of finding the point ϕ^* that maximizes $D_{Q_c}(\phi)$, forming the one parameter family

$$(1 - \gamma)Q_c + \gamma\Delta_{\phi^*}$$

and doing a one-dimensional algorithm in this family (usually Newton–Raphson) to find the maximum along this line. From our earlier theory, we know that this must increase the likelihood. It can be shown that, in fact, it provides a sufficient increase at each step that the algorithm must converge to the NPMLE [Lindsay (1983a)]. (We should more precisely say it has been proven that the likelihood will increase to its maximum value.)

Moreover, this is an algorithm that does not need to keep track of support points because the gradient, the new support point and the new likelihood vector can be constructed from the current likelihood vector. This is fortunate because this algorithm can add a new support point with each iteration.

Intuitive as this may seem, it is generally a very bad idea to use this algorithm except as a supplement to another speedier algorithm. As Lindsay (1983a) indicated, it is sublinear in convergence and in fact becomes slower and slower as we approach convergence.

For this reason, Böhning (1985) suggested a simple alternative that was considerably faster in terms of number of iterations. Called the *vertex exchange method*, it also requires only the solution of a one-dimensional optimization problem at each stage. Although it also has the disadvantage that we must keep track of the current latent distribution, in contrast with the vertex direction method, it can eliminate one support point at each step, so the number of support points stays bounded, with new points replacing old points.

6.3.4. *Intrasimplex direction method.* Neither of the preceding methods could be considered adequately speedy to do many calculations of the NPMLE, such as we might need to construct profile likelihoods or do simulation studies. The problem is that we need to somehow use the fact that unicomponent models are highly correlated when their parameter values are similar, and we cannot take advantage of this multicollinearity without using methods of higher dimension.

For this reason Lesperance and Kalbfleisch (1992) suggested an algorithm of a multivariate type, in which one found the set of optimal weights for a convex combination of the current likelihood vector and the unicomponent likelihood vectors corresponding to the current local maxima of the gradient function. As we have remarked earlier, the log likelihood will be strictly concave in these weight parameters, so a reasonable quasi-Newton procedure to solve for optimal weights, subject to nonnegativity constraints, could be expected to be fast and reliable. In addition, this algorithm does not require storage of the current latent distribution (the number of support points could grow explosively), but just the fitted likelihood vector.

6.3.5. *Monotonicity.* An important piece of practical advice to students starting work in this area is that no matter what method is used, one should be monitoring the likelihood function. If one is working with a method guaranteed to increase the likelihood, such as the EM, then this is a check on the program; if not, it is a preventative for oscillatory behavior. Since there is just a single unique maximum, there is no advantage to having the algorithm

search the space more thoroughly. Böhning, Schlattman and Lindsay (1992) have some suggestions, and the corresponding C.A.MAN program provides various step length options.

6.3.6. *Using the dual problem.* Yet another approach to finding the NPMLE is to switch to solving the dual problem, which is a straightforward optimization problem with linear constraints defining the feasible region. Lesperance and Kalbfleisch (1992) use a canned program for this optimization problem (SIP = semiinfinite programming) and found that it was quite competitive with their intrasimplex direction method. It is the author's intuition that this will often be the best approach to the problem. Table 6.1 reproduces a comparison made by Lesperance and Kalbleish. The "Sup Grad" column indicates the final calculation of the convergence criterion and the column "Δl" indicates the difference between the global maximum log likelihood and the likelihood at the last iteration of the algorithm. These columns will be discussed further when we discuss stopping rules.

6.4. Ideal stopping rules. One could obviously take the tack that one should iterate on an algorithm until the accuracy of the result approaches the limit of machine accuracy. However, this is not practical if one wants to do simulation studies or apply bootstrap methods. We therefore attempt now to describe how to develop statistically meaningful stopping rules.

6.4.1. *The ideal rule.* We postulate that the *ideal stopping rule* for the iterations of an algorithm is to stop when we have a log likelihood that is sufficiently close to the final log likelihood. That is, we quit when we have found Q_{stop} satisfying

$$\ln(L(\hat{Q})) - \ln(L(Q_{\text{stop}})) \leq \text{tol}. \tag{6.1}$$

Here the criterion tol should be related to desired inferential goals. We here suggest what we think are reasonable values based on certain heuristics.

First, we note that setting such a goal will ensure that Q_{stop} will be a consistent estimator of Q, as was shown by Kiefer and Wolfowitz (1956).

Owen (1988) showed that one can construct valid nonparametric confidence sets for the smooth functionals of an unknown distribution function F by examining a form of nonparametric profile likelihood called the empirical

TABLE 6.1
A comparison of algorithms

Algorithm	# Iterations	Sup Grad	Δl	APL Time
VDM	2,177	2.64×10^{-3}	1.07×10^{-3}	12:28
VEM	143	3.18×10^{-2}	0.85×10^{-2}	1:04
ModVDM	60	4.96×10^{-4}	4.98×10^{-4}	5:49
SIP	11	1.60×10^{-8}	0	[0:14]
ISDM	11	1.44×10^{-8}	0	0:03

likelihood. Moreover, the empirical likelihood can be calibrated by the usual chi-squared distribution with degrees of freedom equal to the number of functionals under consideration.

It has not been shown that this theory carries over to our situation, but it seems likely to be true. Even if not, it gives us some guidance as to orders of magnitude that might be relevant. In our setting this would work as follows. Let $\theta = \theta(Q)$ be a functional of Q of interest, such as the mean or the distribution function evaluated at a particular point. Let $\hat{\theta} = \theta(\hat{Q})$. Let $\hat{Q}_\theta$ be the maximum likelihood estimator of Q among all Q that satisfy $\theta(Q) = \theta$. (We will discuss the solution of this optimization problem in the next chapter.) If Owen's result were to hold here, we could say, approximately, that

$$2[\ln(L(\hat{Q})) - \ln(L(\hat{Q}_\theta))] \approx (\hat{\theta} - \theta)[\operatorname{Var}\hat{\theta}]^{-1}(\hat{\theta} - \theta) \tag{6.2}$$

in the neighborhood of the maximum.

Now, suppose we set as a *target of accuracy* that our estimator of θ be within 0.1 standard error of $\hat{\theta}$. This is just 1/40 of the width of a standard confidence interval and so we believe it would be pointless to pursue numerical accuracy further than this, given the statistical inaccuracy. If we set

$$\text{tol} = 0.005, \tag{6.3}$$

stop at Q_{stop} and let θ_{stop} be the value of $\theta(Q_{\text{stop}})$, then

$$2[\ln(L(\hat{Q})) - \ln(L(\hat{Q}_{\theta_{\text{stop}}}))] \leq 2[\ln(L(\hat{Q})) - \ln(L(Q_{\text{stop}}))] \leq 0.01.$$

It follows, given our approximation (6.2), that θ_{stop} deviates from $\hat{\theta}$ by at most 0.1 standard units.

Thus we believe tol = 0.005 is a meaningful statistical goal and going beyond it pursues statistically meaningless accuracy.

6.4.2. *A gradient-based rule.* Now, we obviously cannot at any stage in an algorithm know exactly whether we have met the ideal stopping rule (6.1). However, we remind the reader that we can use the gradient to bound such a difference. That is, we know from Section 5.3.2 that

$$[\ln(L(\hat{Q})) - \ln(L(Q))] \leq \sup_{\phi \in \Omega} D_Q(\phi). \tag{6.4}$$

Thus if we can ensure that

$$\sup_{\phi \in \Omega} D_Q(\phi) \leq \text{tol},$$

we would have our targeted statistical accuracy.

One important question here regards whether the inequality (6.4) is close to an equality, at least in order of magnitude; otherwise we may be pushing our accuracy goals substantially beyond that actually needed. One piece of evidence for this is in Lesperance and Kalbfleisch (1992), in a table reproduced as Table 6.1 above. Our interest is in comparing columns 3 and 4, where we find the sup gradient and remaining likelihood to be gained at the final

steps of each of the algorithms. We note that the sup gradient upper bound result shows that the last two algorithms have converged to a high degree of accuracy, so their likelihoods were set to be the true maximum. The numbers for the other estimators suggest that the order of magnitude of the remaining likelihood increase can be predicted fairly well from the sup gradient. In the worst case, the sup gradient was four times the remaining likelihood increase.

6.4.3. *Combining grid and gradient.* We now return to the supposition that we will evaluate the gradient at a finite grid of points. The next question we ask is: can we select a grid Ω_s and a modified tolerance level tol* in such a way that

$$\sup_{\phi\in\Omega_s} D_Q(\phi) \le \text{tol}^* \quad \Longrightarrow \quad \sup_{\phi\in\Omega} D_Q(\phi) \le \text{tol}? \tag{6.5}$$

Such a bound cannot be created unless the gradient has boundable variation, so that knowledge of its values on the grid points determines how high it can go between grid points.

This is not true in some cases, such as the empirical CDF problem, where one must include all the data points in the grid or else have zero likelihood for the data. In fact, as we will see, the larger the second derivatives of the unicomponent density are, the more refined must the grid be to attain desired statistical accuracy.

To analyze this question, we restrict attention to real-valued ϕ. Suppose that Q satisfies

$$\sup_{\phi\in\Omega_s} D_Q(\phi) \le \text{tol}^*$$

and suppose that g_1 and g_2 are two adjoining grid points. Suppose that we can construct a bound of the type

$$\inf_{\phi\in[g_1,g_2]} D_Q''(\phi) \ge -c. \tag{6.6}$$

This ensures that the gradient cannot go upward and then curve downward too fast, and so bounds the maximum via

$$\sup_{\phi\in[g_1,g_2]} D_Q(\phi) \le \text{tol}^* + \frac{c}{2}\left(\frac{g_2-g_1}{2}\right)^2. \tag{6.7}$$

(The reader is invited to check this out: The bound arises by setting the gradient equal to tol* at the two endpoints and then making the steepest possible quadratic in between.)

Thus, it is clear that if we can find c to satisfy (6.6), then by application of (6.7), together with a fine enough grid, one can attain the goal (6.5).

6.4.4. *Bounding the second order score.* One of the problems with establishing a bound of the type (6.6) is the presence of the arbitrary distribution Q as an argument. Thus we will modify the above strategy somewhat to make the problem easier to solve. We remind the reader that if $L_i(\phi) = f(x_i;\phi)$, then the dispersion score

$$v_2(\phi, x_i) = \frac{f''(x_i;\phi)}{f(x_i;\phi)} = \frac{L_i''(\phi)}{L_i(\phi)}$$

played an important role in evaluating overdispersion.

The following lemma suggests that this score is an important quantity in determining the properties of the gradient as well.

LEMMA 26. *If for all i and for all $\phi \in [g_1, g_2]$,*

$$\frac{L_i''(\phi)}{L_i(\phi)} \geq -k, \tag{6.8}$$

then

$$D_Q''(\phi) \geq -k\left[n + \sup_{\phi\in[g_1,g_2]} D_Q(\phi)\right]. \tag{6.9}$$

PROOF. Straightforward algebra, using the fact that

$$n + D_Q(\phi) = \sum n_i \frac{L_i(\phi)}{L_i(Q)}. \qquad \square$$

Before proceeding, we note that it is easy to establish a bound such as (6.8) in the exponential family. For example, if ϕ is the natural parameter, we have

$$\frac{L_i''(\phi)}{L_i(\phi)} = v_2(\phi, x_i) = [x_i - \mu(\phi)]^2 - \sigma^2(\phi),$$

which is clearly bounded below by $-\sigma^2(\phi)$ on the chosen interval. If we are using instead the mean value parameterization, then

$$\frac{L_i''(\phi)}{L_i(\phi)} = \frac{(x_i - \mu(\phi))^2 - \sigma^2(\phi)}{\sigma^4(\phi)} \geq -\frac{1}{\sigma^2(\phi)}$$

is a simple bound on an interval in which the unicomponent variance of X does not go to zero.

6.4.5. *A conservative method.* Now if we insert (6.9) into (6.5), we get

$$\sup_{\phi\in[g_1,g_2]} D_Q(\phi) \leq \text{tol}^* + k\left(\frac{(g_2 - g_1)^2}{8}\right)\left[n + \sup_{\phi\in[g_1,g_2]} D_Q(\phi)\right].$$

We let

$$C = k\left(\frac{(g_2 - g_1)^2}{4}\right)$$

be the *critical factor* for the grid. For example, if we use the mean value parameter of the exponential family, we get

$$C = \frac{w^2}{\sigma^{*2}},$$

where w is the half-width of the grid separation and σ^{*2} is a lower bound for $\sigma^2(\phi)$ on that interval. Algebraic manipulation then gives the upper bound

$$\sup_{\phi \in [g_1, g_2]} D_Q(\phi) \leq \frac{\text{tol}^* + Cn/2}{1 + C/2}.$$

Clearly for C sufficiently small, one can make this bound as close to tol* as one likes. Moreover, for C small, the bound is approximately $\text{tol}^* + Cn/2$.

We note that the choice of an acceptable critical factor C for the grid will therefore depend very much on the sample size n. This derives from the fact that as the sample size increases, the greater the precision of our statistical knowledge and so the greater the need for numerical accuracy. Thus we run into the unfortunate side effect that *the larger the sample size, the more difficult the numerical problem.*

Returning to the exponential family in the mean value parameterization, we see that the critical factor for the grid is the grid separation expressed in standard deviation units. Thus, for example, if we were to separate the grid points by 0.02 standard deviations, we would get an upper bound of the form

$$\sup_{\phi \in \Omega} D_Q(\phi) \leq \frac{\text{tol}^* + (0.00005)n}{1 + (0.00005)}.$$

In this case, the standardized grid widths must shrink at the rate $n^{-1/2}$ to maintain the desired accuracy. For example, if our target tolerance is 0.005, as suggested earlier, and we set tol* for the grid at 0.0025, then we must set

$$\frac{w}{\sigma^*} = \sqrt{0.005} n^{-1/2}.$$

Thus for $n = 100$, we need a standardized half-grid-width of about 0.007. For $n = 10{,}000$, the grid half-width shrinks to 0.0007 standardized units.

6.4.6. *Remarks.* We remark that this analysis indicates that if we were to change to an exponential family parameterization in which the variance is constant, then we could use an equally spaced grid without losing accuracy in any region, but otherwise there will be a potential loss of information in the use of an equally spaced grid.

These design considerations have been based on the idea of conservatism and least favorable situations. Empirical evidence is not available on whether these recommendations are overly conservative.

We also note that if one is doing a gradient search based on a grid of starting points, that these considerations suggest that the grid should be evenly spaced on the standardized scale, and that the spacing should shrink with sample size.