THE UNIVERSITY OF CHICAGO

CONSTRUCTION, IMPLEMENTATION, AND THEORY OF ALGORITHMS BASED ON DATA AUGMENTATION AND MODEL REDUCTION

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF STATISTICS

BY

DAVID ANTHONY VAN DYK

CHICAGO, ILLINOIS

AUGUST 1995

Acknowledgements

I wish to thank my advisor, Xiao-Li Meng, for his unending support as I completed this project. He is a gifted and patient teacher who never failed to make time to answer my questions, to point me in the direction of interesting and fruitful research topics, or to think carefully and critically about my work. I am very appreciative of his willingness to go far beyond the call of duty.

There are many others who have helped greatly in the completion of this thesis. Here I can mention only a few, but would like to thank Don Rubin for his help with Chapter 4, Augustine Kong for helping me obtain two summers of research support, Jeffrey Fessler for comments pertaining to the SAGE algorithm and Chapter 5, Sam Vandervelde for his never-ending willingness to help me with the subtler details of the mathematics underlying my work, and Ron Thisted for comments on presentation and pointing out several helpful references.

This work was supported in part by National Science Foundation grant DMS 89-05292, the Department of Education (through GAANN program awards P200A10027 and P200A40313), the U.S. Public Health Service/National Institutes of Health (PHS/NIH GM 46800), The University of Chicago Louis Block Fund, and the U.S. Census Bureau through a contract with the National Opinion Research Center at the University of Chicago. Computations for this document were

performed using computer facilities supported in part by the National Science Foundation under grants DMS 89-05292 and DMS 87-03942 awarded to the Department of Statistics at The University of Chicago, and by The University of Chicago Block Fund. I am grateful to all of these sources for their financial support.

Finally, I wish to thank Greg Jao and Peter Vassilatos for their many editorial comments.

Abstract

In the thesis we provide a general framework for maximum likelihood algorithms based on data augmentation and model reduction. Starting with this theoretical framework, we explore methods of constructing and implementing efficient algorithms. We show how to derive faster algorithms by optimizing the rate of convergence as a function of a working parameter which is introduced into the data-augmentation scheme. We then propose the Alternating Expectation/Conditional Maximization or AECM algorithm which includes the EM, ECM, ECME, and SAGE algorithms as special cases. We also show how the matrix rate of convergence can be used to compute the asymptotic variance-covariance matrix of the maximum likelihood estimates. The relative efficiency of competing model-reduction schemes is explored via permutation of the conditional maximization steps within the ECM algorithm. Finally, we explore the inferential use of the data-augmentation scheme in the context of estimating the number of components in a finite mixture, with possible extensions to other model-fitting problems.

Chapter 1

EM-type Algorithms: Background and Notation

1.1. A Brief Overview

The Expectation/Maximization or EM algorithm (Dempster, Laird, and Rubin, 1977) is a formalization of an old ad hoc method for handling missing data. If we had observed the missing values, we could estimate the parameters of a posited model using standard complete-data techniques. On the other hand, if we knew the model parameters, we could impute the missing data according to the model. This leads naturally to an iterative scheme. The advantage of the EM formulation over its ad hoc predecessor is that it recognizes that the correct imputation is through the complete-data sufficient statistics, or more generally through the complete-data loglikelihood function, and not the individual missing values. Specifically, at each iteration the E-step computes the conditional expectation of the complete-data log-likelihood given the observed data and the previous iterate of the parameter value, and the M-step then maximizes this imputed loglikelihood function to determine

the next iterate of the parameter. We repeat this process until the algorithm converges. Since EM separates the complete-data analysis from the extra complications due to missing data and allows the use of complete-data maximization techniques, it is both conceptually and computationally simple. When facing an incomplete-data problem, we can first ask what would be done if there were no missing values, and then proceed with the help of EM to deal with the missing data, assuming that the missing-data mechanism (Rubin, 1976) has been taken into account. This advantage has helped EM win great popularity among practical users. Meng and Pedlow's (1992) bibliography reveals that there are more than 1,000 EM-related articles in almost 300 journals, most of which are outside the field of statistics.

1.1.1. Model reduction

In some cases, the complete-data problem itself may be complicated. For instance, when a model has many parameters, finding maximum likelihood estimates (MLEs) can be a demanding task. A natural strategy, in general, is to break a big problem into several smaller ones. If some of the model parameters were known, it might be easier to estimate the rest. In the complete-data problem, we can partition the parameters into several sets and estimate one set conditional on all the others. This model-reduction technique is well-known in the numerical analysis literature as the cyclic coordinate ascent method (e.g. Zangwill, 1969) and is called in statistical terms the Conditional Maximization or CM algorithm by Meng and Rubin (1993), whose model-reduction scheme goes beyond a simple partition of the parameter, as they find a more sophisticated model-reduction scheme is useful for certain statisti-

cal models. The Expectation/Conditional Maximization or ECM algorithm (Meng and Rubin, 1993) is an efficient combination of the CM and EM algorithms. It replaces the maximization step of EM with a set of conditional maximization steps, and thus splits a difficult maximization problem into several easier ones. Consequently, in many practical applications, the use of model reduction in ECM extends the flexibility and power of EM while retaining its stability in the sense of monotonic convergence of the likelihood along the induced sequence to the MLE. The flexibility introduced by model reduction also allows more efficient data-augmentation schemes, as we shall explore in this thesis.

1.1.2. Data-augmentation

Although the principal reasons for the popularity of EM and ECM are their simplicity and stability, they are sometimes criticized because of their slow convergence in some applications. Loosely speaking, the rate of convergence is governed by the amount of missing information in terms of the observed Fisher information. The more missing information, the slower the algorithms will converge. Although EM is motivated by the idea of a missing-data structure, in many of its novel applications, there is strictly speaking no missing data in the usual sense. That is, the observed data is simply augmented to some larger data set for which analysis is simpler (for this reason, in what follows we will use the more general term "augmented data" in place of "complete data"). Choosing a sensible data-augmentation scheme is an art which requires compromising between simplicity (which often means more augmentation) and fast convergence (which often requires less augmentation). Thus,

careful selection of the data-augmentation scheme can lead to simpler and faster algorithms. Good examples of careful selection include, the Expectation/Conditional Maximization Either or ECME algorithm (Liu and Rubin, 1995a) and the Space-Alternating Generalized EM or SAGE algorithm (Fessler and Hero, 1994), both of which are extensions of the ECM algorithm. Both algorithms incorporate effective data-augmentation schemes to improve the rate of convergence of the algorithm. A primary contribution of this thesis is to illustrate a new technique for the construction of effective data-augmentation schemes, which leads to algorithms that not only maintain the simplicity and stability of the EM algorithm but also substantially improve upon its rate of convergence. In the problems we consider, the new algorithms are often ten times or even hundreds of times faster than their standard counterparts in terms of actual computation time.

1.1.3. Synopsis

In what follows, we will explore methods of constructing, implementing, and analyzing algorithms which incorporate both model-reduction and effective data augmentation into the EM algorithm. Our exploration also illustrates the theoretical, computational, and inferential use of the rate of convergence of EM-type algorithms. Chapter 2 focuses on constructing optimal algorithms in terms of their rate of convergence via the introduction of a working parameter indexing a class of data-augmentation schemes. As we will see, this leads to simple changes in some standard algorithms and results in dramatic increases in computational efficiency. In Chapter 3 we will develop the Alternating Expectation/Conditional Maximiza-

tion or AECM algorithm, a generalization of EM which incorporates both model reduction to simplify implementation and a scheme that allows the data augmentation to be altered at each iteration to improve the overall rate of convergence of the algorithm. AECM will be shown to include not only EM, ECM, and SAGE but also the soon-to-be-discussed PECM and MCECM algorithms (Meng and Rubin, 1993) as well as the special case of ECME for which Liu and Rubin's (1995a) convergence theorems hold; we will discuss why their general theory applies only to this special case.

All of these algorithms are designed to calculate maximum likelihood estimates or posterior modes. In most statistical analysis, however, measures of uncertainty (e.g., asymptotic variance-covariance matrix of the estimates) are also needed. Chapter 4 develops the Supplemented ECM or SECM algorithm which is designed to do this when implementing ECM and which can be used in most implementations of the AECM algorithm. As we have seen, we incorporate model reduction into EM by breaking the maximization step into several conditional maximization steps. The order that these steps are performed is trivial to change but generally affects the performance of the algorithm (e.g., rate of convergence). Chapter 5 thus explores the effect of permutation of conditional maximization steps in the context of ECM and illustrates several valuable lessons pertaining to the incongruence of empirical and theoretical results that will have implications in other studies. Finally, Chapter 6 looks at the inferential use of the data-augmentation scheme through the rate of convergence of EM in the context of estimating the number of components in a finite mixture, with possible extensions to other model fitting problems.

The remainder of the current chapter outlines the details and notation of the EM, ECM, ECME and SAGE algorithms, as well as the theory of the rate of convergence of EM-type algorithms, thereby explicitly illustrating model reduction and data augmentation in EM-type algorithms.

1.2. The EM Algorithm

Let $L(\theta|Y_{\rm obs}) = \log f(Y_{\rm obs}|\theta)$ be the observed-data loglikelihood function that we want to maximize, where $\theta = (\theta_1, \ldots, \theta_d)$ is a d-dimensional model parameter with domain Θ. (For simplicity, we assume this model already has incorporated any non-ignorable missing-data mechanism; see Rubin, 1976.) Let $f(Y_{\rm aug}|\theta)$ be a density for the augmented data $Y_{\rm aug} = (Y_{\rm obs}, Y_{\rm mis})$, where $Y_{\rm mis}$ is the missing (i.e., unobserved) part. The augmented data are chosen such that maximizing $L(\theta|Y_{\rm aug}) = \log f(Y_{\rm aug}|\theta)$ is much easier than directly maximizing $L(\theta|Y_{\rm obs})$. This is the setting in which EM and its extensions are most useful.

Starting with an initial value $\theta^{(0)} \in \Theta$, the EM algorithm finds $\theta^\star$, a maximizer of $L(\theta|Y_{\rm obs})$, by iterating the following two steps $(t = 0, 1, \ldots)$:

E-step: Impute the unknown augmented-data loglikelihood $L(\theta|Y_{\rm aug})$ by its conditional expectation given $Y_{\rm obs}$ and the current estimate $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = \int L(\theta|Y_{\rm aug}) f(Y_{\rm mis}|Y_{\rm obs}, \theta^{(t)}) {\rm d}Y_{\rm mis}. \qquad (\mathbf{1.2.1})$$

M-step: Determine $\theta^{(t+1)}$ by maximizing the imputed loglikelihood $Q(\theta|\theta^{(t)})$:

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \quad \text{for all } \theta \in \Theta. \qquad (\mathbf{1.2.2})$$

For exponential families, $Q(\theta|\theta^{(t)}) = L(\theta|S^{(t)}(Y_{\text{obs}}))$, where $S^{(t)}(Y_{\text{obs}}) = \mathsf{E}[S(Y_{\text{aug}})|Y_{\text{obs}}, \theta^{(t)}]$ with $S(Y_{\text{aug}})$ being the augmented-data (vector) sufficient statistic. The E-step therefore reduces to finding the conditional expectation of $S(Y_{\text{aug}})$, and maximizing $Q(\theta|\theta^{(t)})$ is computationally the same as maximizing $L(\theta|Y_{\text{aug}})$, the augmented-data loglikelihood. The latter is one of the principal reasons for the popularity of the EM algorithm in practice because it allows practitioners to use existing (complete-data) techniques and software when Y_{aug} is properly chosen.

The convergence properties of EM were established by Dempster, Laird, and Rubin (1977) and Wu (1983). In particular, EM is a GEM (Generalized EM) which ensures that $L(\theta^{(t+1)}|Y_{\text{obs}}) \geq L(\theta^{(t)}|Y_{\text{obs}})$ for any sequence $\{\theta^{(t)} : t \geq 0\}$ of EM iterates. Moreover, given mild regularity conditions, it can be shown that EM converges to a stationary point (typically a local mode in practice) of $L(\theta|Y_{\text{obs}})$. These stability properties combined with simple implementation are very attractive to analysts who are not necessarily numerically sophisticated and whose main objectives are not computational or numerical. The following recent generalizations of EM are aimed to further enhance the applicability, as well as efficiency, of EM in practice.

1.3. The ECM Algorithm

In some applications of EM, the M-step many not be in closed form, in which case EM looses its simplicity because it requires nested iterations within each M-step. In many such cases, the ECM algorithm, which replaces the maximization of $Q(\theta|\theta^{(t)})$ by several simpler conditional maximizations, can regain the simplicity of EM. Specifically, let $G = \{g_s(\theta), s = 1, \ldots, S\}$ be a set of $S \geq 1$ preselected vector functions that are "space filling" (Meng and Rubin, 1993) in the sense of allowing maximization over the full space Θ. ECM incorporates the model reduction determined by G into the M-step by replacing it with S Conditional Maximization (CM) steps:

s^{th} *CM-step:* Find $\theta^{(t+\frac{s}{S})}$ such that

$$Q(\theta^{(t+\frac{s}{S})}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \quad \text{for all } \theta \in \Theta_s^{(t)} \equiv \{\theta \in \Theta : g_s(\theta) = g_s(\theta^{(t+\frac{(s-1)}{S})})\}, \tag{1.3.1}$$

where $s = 1, \ldots, S$, and the next iterate $\theta^{(t+1)} \equiv \theta^{(t+\frac{S}{S})}$. The rationale behind the CM-steps is that in problems where maximizing $Q(\theta|\theta^{(t)})$ over $\theta \in \Theta$ is difficult, it may be possible to choose G so that it is simple to maximize over $\theta \in \Theta_s^{(t)}$ for $s = 1, \ldots, S$.

For example, a common useful choice of G is to choose $g_s(\theta) = (\vartheta_1, \ldots, \vartheta_{s-1}, \vartheta_{s+1}, \ldots, \vartheta_S)$ for $s = 1, \ldots, S$, where $(\vartheta_1, \ldots, \vartheta_S)$ is a partition of θ. In other words, at the s^{th} CM step, we maximize $Q(\theta|\theta^{(t)})$ over ϑ_s with the rest of the $S - 1$ subvectors fixed at their previous estimates. This common special class of ECM is called the partitioned ECM or PECM algorithm by Meng and Rubin

(1992). More complicated choices of G can also be useful in practice, as we will illustrate in Section 4.4.4.

A slight modification of ECM can improve its speed in some settings. The multi-cycle ECM or MCECM algorithm (Meng and Rubin, 1993) is a variation in which extra E-steps are added to each iteration in the hope of speeding up the convergence. Consider, for example, the three CM-step ECM algorithm, ECM : $\text{E} \rightarrow \text{CM}_1 \rightarrow \text{CM}_2 \rightarrow \text{CM}_3$. The MCECM algorithm adds one or more E-step to each iteration, for example,

$$\text{MCECM}: \quad \text{E} \rightarrow \text{CM}_1 \rightarrow \text{E} \rightarrow \text{CM}_2 \rightarrow \text{E} \rightarrow \text{CM}_3. \qquad (\mathbf{1.3.2})$$

In the MCECM algorithm, each of the E-steps are computed the same way as in (1.2.1) with $\theta^{(t)}$ being the most up-to-date iterate of θ.

The convergence properties of ECM and its variations were established in Meng and Rubin (1993) and are almost identical to those of EM presented in Dempster, Laird and Rubin (1977) and Wu (1983). In particular, for any ECM (or MCECM) sequence $\{\theta^{(t)}, t = 0, 1, \ldots\}$, $L(\theta^{(t+1)}|Y_{\text{obs}}) \geq L(\theta^{(t)}|Y_{\text{obs}})$, that is, at each iteration an ECM sequence increases the likelihood being maximized.

1.4. The ECME Algorithm

The ECM algorithm generalizes EM by incorporating model reduction into the M-step in order to regain the simplicity of EM. As expected, replacing the M-step by a sequence of CM-steps can slow down convergence. (Surprisingly, this is not universally true; see the counter-example provided by Meng, 1994.) Both the ECME and SAGE algorithms use creative data-augmentation schemes to improve the speed of convergence, and interestingly, such improvement is possible because of the flexibility introduced by the model reduction (i.e., we can now use several different data-augmentation schemes because the model has been broken-up into several parts).

In their development of ECME, Liu and Rubin (1995a) recognize that in some applications of the ECM algorithm the implementation of some CM-steps requires similar computations for maximizing the conditional observed-data likelihood and for conditional augmented-data likelihood, and thus, it is computationally more efficient to directly maximize the former. That is to say that, motivated by the principle that augmenting less results in faster algorithms, we can improve the speed of convergence, often substantially, by not augmenting at all in some of the CM-steps, provided we do not increase the complexity of implementing these CM-steps. (Such a strategy, i.e., not augmenting, is not useful in the original EM implementation because it eliminates the EM algorithm altogether.)

In the ECME algorithm presented in Liu and Rubin (1995a), any of the CM-steps may be chosen to act on $L(\theta|Y_{\text{obs}})$ instead of $Q(\theta|\theta^{(t)})$. Unfortunately,

their proofs of the convergence results contain an error, as shall be discussed in Chapter 3. We, therefore, present a somewhat restricted version of ECME whose convergence will be proven as a special case of the AECM algorithm in Section 3.3. Specifically, we require that at every iteration, the CM-steps which act on $Q(\theta|\theta^{(t)})$ all be performed *before* those which act on $L(\theta|Y_{\text{obs}})$. That is, for $S_0 \leq s \leq S$, the CM-step given in (1.3.1) is replaced by

s^{th} *CM-step:* Find $\theta^{(t+\frac{s}{S})}$ such that

$$L(\theta^{(t+\frac{s}{S})}|Y_{\text{obs}}) \geq L(\theta|Y_{\text{obs}}), \quad \text{for all } \theta \in \Theta_s^{(t)} \equiv \{\theta \in \Theta : g_s(\theta) = g_s(\theta^{(t+\frac{(s-1)}{S})})\}. \tag{1.4.1}$$

The first $S_0 - 1$ CM-steps remain as in ECM.

Liu and Rubin (1995a) give several examples in which the increased computation and/or human effort required by the constrained maximization of $L(\theta|Y_{\text{obs}})$ is greatly outweighed by the improved rate of convergence of the algorithm, with substantial savings of actual computer time.

1.5. The SAGE Algorithm

Like the ECME algorithm, the SAGE algorithm (Fessler and Hero, 1994) is designed to speed up the convergence of EM. Although Fessler and Hero developed the SAGE algorithm without knowledge of ECM, their algorithm is easily understood as a generalization of a multi-cycle PECM algorithm. That is, we will start with a MCECM algorithm in which *each* CM-step is preceded by an E-step (i.e. (1.3.2))

and the constraint functions which define the CM-steps are of the special form which partitions the parameter space as in PECM. Fessler and Hero (1994) recognized not only that less data-augmentation results in faster EM-type algorithms but also that a different data-augmentation scheme can be used in each E-step/CM-step pair. This is illustrated in the SAGE algorithm which, at iteration $t+1$, partitions the (unordered) parameter θ into an active component ϑ_{t+1} and a fixed component φ_{t+1} and chooses the data augmentation $Y_{\text{aug}}^{(t+1)}$ to be used in the iteration:

E-step: Compute

$$Q_{t+1}(\theta|\theta^{(t)}) = \int L(\theta|Y_{\text{aug}}^{(t+1)}) f(Y_{\text{mis}}^{(t+1)}|Y_{\text{obs}}, \theta^{(t)}) \mathrm{d}Y_{\text{mis}}^{(t+1)},$$

where $Y_{\text{aug}}^{(t+1)} = (Y_{\text{obs}}, Y_{\text{mis}}^{(t+1)})$.

CM-step: Determine $\theta^{(t+1)}$ by maximizing $Q_{t+1}(\theta|\theta^{(t)})$ under the constraint $\varphi_{t+1} = \varphi_{t+1}^{(t)}$:

$$Q_{t+1}(\theta^{(t+1)}|\theta^{(t)}) \geq Q_{t+1}(\theta|\theta^{(t)})$$

for all θ such that $\varphi_{t+1} = \varphi_{t+1}^{(t)}$.

Clearly the sequence $\{\vartheta_t, t \geq 1\}$ must be chosen carefully so that the resulting algorithm maximizes over all of Θ, which we will formalize in Chapter 3. Note that iterations are counted differently in SAGE by Fessler and Hero (1994) then in ECM or MCECM. A SAGE iteration consists of one CM-step along with its E-step, whereas a (MC)ECM iteration consists of a space-filling set of CM-steps along with the E-step(s).

Like EM, the SAGE algorithm increases $L(\theta|Y_{\text{obs}})$ at each iteration (Fessler and Hero, 1994) and, as we shall prove in Section 3.3, converges to a stationary point of $L(\theta|Y_{\text{obs}})$ under mild regularity conditions. The advantage of SAGE is its allowance of adaptive data augmentation, thus improving the speed of the algorithm. In the context of medical imaging, Fessler and Hero (1994) provide both theory and examples of the faster convergence of SAGE. The ECME algorithm also can be viewed as a special case of SAGE (when ECM is a two-CM-step PECM) in the sense that some CM-steps require no augmentation.

1.6. The Rate of Convergence of EM-type Algorithms

Like any deterministic iterative algorithm, an EM-type algorithm implicitly defines a mapping $M : \theta^{(t)} \rightarrow \theta^{(t+1)} = M(\theta^{(t)})$ from the parameter space Θ to itself. Suppose that $M(\theta)$ is differentiable in a neighborhood of $\theta^{\star}$, then a Taylor's series approximation yields

$$(\theta^{(t+1)} - \theta^{\star}) \approx (\theta^{(t)} - \theta^{\star}) DM(\theta^{\star}) \qquad (\mathbf{1.6.1})$$

where

$$DM(\theta) = \left(\frac{\partial M_j(\theta)}{\partial \theta_i} \right).$$

When $DM(\theta^{\star})$ is nonzero, which is the case for EM-type algorithms, the mapping is linear if we ignore the higher order terms in the Taylor's series expansion. This approximation becomes exact at convergence of the algorithm, and thus, $DM(\theta^{\star})$ is

called the (matrix) rate of convergence (e.g., Meng, 1994). In what follows $DM(\theta^\star)$ will always be evaluated at $\theta = \theta^\star$. Thus, we will suppress its dependency on θ.

For the EM algorithm, Dempster, Laird, and Rubin (1977) established the following fundamental identity. Suppose $Q(\theta|\theta^{(t)})$ is maximized by setting its first derivative equal to zero, and $\theta^\star$ is in the interior of Θ. Then, the matrix rate of EM is given by

$$DM^{EM} = I_{\text{mis}} I_{\text{aug}}^{-1} = I_d - I_{\text{obs}} I_{\text{aug}}^{-1}, \tag{**1.6.2**}$$

where

$$I_{\text{mis}} = \int -\frac{\partial^2 \log f(Y_{\text{mis}}|Y_{\text{obs}}, \theta)}{\partial\theta \cdot \partial\theta^\top} f(Y_{\text{mis}}|Y_{\text{obs}}, \theta) dY_{\text{mis}} \Big|_{\theta=\theta^\star} \tag{**1.6.3**}$$

is the expected missing information,

$$I_{\text{aug}} = \int -\frac{\partial^2 \log f(Y_{\text{aug}}|\theta)}{\partial\theta \cdot \partial\theta^\top} f(Y_{\text{mis}}|Y_{\text{obs}}, \theta) dY_{\text{mis}} \Big|_{\theta=\theta^\star} \tag{**1.6.4**}$$

is the expected augmented information,

$$I_{\text{obs}} = I_o(\theta^\star|Y_{\text{obs}}) = -\frac{\partial^2 L(\theta|Y_{\text{obs}})}{\partial\theta \cdot \partial\theta} \Big|_{\theta=\theta^\star} \tag{**1.6.5**}$$

is the observed information matrix, and I_d is a $d \times d$ identity matrix. Identity (1.6.2) is fundamental because it directly relates the rate of convergence of EM with the matrix fraction of missing information, $I_{\text{mis}} I_{\text{aug}}^{-1}$. If the augmented information is large relative to the observed information, DM^{EM} will be close to the identity and EM will converge slowly. On the other hand, if the augmented information is nearly equal to the observed information, DM^{EM} will be near zero, and EM will converge quickly. Identity (1.6.2) is also crucial to the Supplemented EM or

SEM algorithm (Meng and Rubin, 1991a), which computes the asymptotic variance-covariance matrix of $\theta^\star$, namely I_{obs}^{-1}, when implementing EM.

The matrix rate of convergence for the ECM algorithm (Meng, 1994) can be expressed as a product of the matrix rates for the EM and CM algorithms

$$DM^{ECM} = DM^{EM} + [I_d - DM^{EM}]DM^{CM}, \tag{1.6.6}$$

where DM^{CM} is the matrix rate of the CM algorithm and is given by

$$DM^{CM} = P_1 \cdots P_S, \tag{1.6.7}$$

with

$$P_s = \nabla_s[\nabla_s^\top I_{\text{aug}}^{-1}\nabla_s]^{-1}\nabla_s^\top I_{\text{aug}}^{-1}, \qquad s = 1, \ldots, S \tag{1.6.8}$$

and $\nabla_s = \nabla g_s(\theta^\star)$ being the gradient of the constraint function $g_s(\theta)$ evaluated at $\theta = \theta^\star$. In Chapter 4, we will use (1.6.6) to develop the SECM algorithm which calculates the asymptotic variance-covariance matrix of $\theta^\star$ when implementing ECM.

Although Liu and Rubin (1995a) generalize (1.6.6) to an expression for the matrix rate of ECME, a simpler expression can be derived for the corrected version of ECME described in Section 1.4, as well as for the SAGE algorithm, since these are both instances of AECM algorithms, which will be discussed in Chapter 3.

The global rate of convergence of the EM algorithm is defined as the limit of

$$r_t = \frac{||\theta^{(t)} - \theta^\star||}{||\theta^{(t-1)} - \theta^\star||}, \quad t \geq 1 \tag{1.6.9}$$

as $t \to \infty$, where $||\cdot||$ is the Euclidean norm. Algorithms which have smaller values of r_t tend to converge more quickly. For EM the global rate of convergence,

$r = \lim_{t\to\infty} r_t$, always exists; under certain regularity conditions is equal to the largest eigenvalue of DM^{EM} ; and lies in the unit interval (see Meng and Rubin, 1994a). In practice, an easily computable measure of the global rate of convergence is the empirical rate, $\hat{r} = \lim_{t\to\infty} \hat{r}_t$, where $\hat{r}_t = ||\theta^{(t)} - \theta^{(t-1)}||/||\theta^{(t-1)} - \theta^{(t-2)}||$. In Chapter 2 we will minimize r as a function of a working parameter introduced into the data augmentation, thereby using (1.6.2) to optimize the efficiency of EM. For other algorithms, a more complicated measure of the global rate of convergence may be needed (e.g., the root convergence factor). In Chapter 5 we will generalize the global rate to the ECM algorithm and investigate its usefulness in predicting the actual number of steps required for convergence of ECM.

Chapter 2

Efficient Data Augmentation: The Key to the Rate of Convergence

2.1. Speeding Up EM with Little Sacrifice

Since Dempster, Laird, and Rubin (1977) showed its great practical potential for finding maximum likelihood estimates or posterior modes, the EM algorithm has become one of the most well-known and used techniques in applied statistics. Although the principal reasons for this popularity are its easy implementation and stable convergence, various attempts have been made in the literature to speed up EM as it has been observed that EM can converge slowly since it is a linear iteration (in contrast with the Newton-Raphson algorithm, which converges superlinearly with careful implementation and monitoring). Proposed methods to speed up EM include the use of Aitkin acceleration (e.g., Dempster, Laird and Rubin, 1977; Louis, 1982; Lindstram and Bates, 1988), combining it with Newton-Raphson-type algorithms (e.g., Lange, 1995) or conjugate gradient methods (e.g., Jamshidian and Jennrich, 1993). An undesirable feature of these accelerations is that the savings

in computer time is achieved typically at the expense of a much larger human investment for general users since these methods require not only more numerically complex implementations but also more careful monitoring, and even with such care the algorithms may not converge properly (e.g., Lansky and Casella, 1990).

However, there is a way of improving the speed of EM without much sacrifice of its simplicity or stability. Since the rate of convergence of EM is determined by the fraction of missing information (e.g., (1.6.2)), the data-augmentation scheme one uses for constructing the augmented-data likelihood (or posterior) determines the speed of EM. It has been well recognized since Dempster, Laird and Rubin (1977) that by augmenting less, one can have a faster algorithm, but a common trade-off is that the resulting M-step and/or E-step may be more difficult to implement. If the M-step and E-step resulting from less augmentation are equally simple (or somewhat less simple if the gain in speed is relatively substantial), then there is no reason not to use the faster EM. This is, for example, the motivation and advantage of the ECME algorithm and of the SAGE algorithm described in Chapter 1.

In this chapter we will present an approach that uses this idea for accelerating EM by searching for an efficient data-augmentation scheme. By "efficient" we mean less augmentation while maintaining the simplicity and stability of EM. Previous attempts, as presented in Liu and Rubin (1995a) and Fessler and Hero (1994), have stemmed from comparing several natural data-augmentation schemes inherent in the underlying problems. Our key idea here is to introduce a working parameter to index a class of possible data-augmentation schemes, most of which are not "natural" in the original problem, to facilitate our search. Section 2.2 provides

the necessary theoretical derivations to compare the rates of convergence of EM algorithms resulting form the data-augmentation schemes indexed by the working parameter. In particular, we will show that minimizing the augmentation in terms of the observed Fisher information results in the optimal EM algorithm. In Sections 2.3 we apply this idea to the problem of fitting univariate and multivariate t-models and construct a class of algorithms which includes both the standard EM algorithm and an interesting algorithm proposed in Kent, Tyler, and Vardi (1994), which was up until now not recognized as EM. In Section 2.4 we will present empirical evidence of the improvement of the optimal EM over the standard EM which is particularly substantial (e.g., often more than 10 times faster) for small degrees of freedom and/or large dimension and prove that the optimal EM is faster than (or as fast as) the standard EM for any t-model being fit to any data set (not necessarily from the posited t-model). In Sections 2.5 we apply the same idea to the random-effects model and present several new algorithms along with an empirical comparison (Section 2.6) showing dramatic improvement (e.g. often more than 100 times faster) when the variance due to the random effects does not dominate the residual variance. The idea of introducing a working parameter (or more generally other structures, deterministic or random) into the data-augmentation scheme appears to be very general and powerful, and we hope the work presented here will stimulate further research in this direction, research that has direct practical impact.

2.2. Ordering Data-Augmentation Schemes

In general when constructing an EM algorithm, any data set can be used as the augmented data so long as it contains $Y_{\rm obs}$. Suppose we have a class of augmented-data sets $Y_{\rm aug}(a)$ with a working parameter a contained in an index set $\mathcal{A}$, such that $Y_{\rm aug}(a)$ contains $Y_{\rm obs}$ for each $a \in \mathcal{A}$, our goal is to determine values of a that result in algorithms that are both quick to converge and easy to implement. The question of ease of implementation must be considered on a case-by-case basis, so for the moment we confine our attention to the rate of convergence of EM as a function of a and write both the global rate, $r(a)$, and the matrix rate, $DM^{EM}(a) = I - I_{\rm obs}I^{-1}_{\rm aug}(a)$, as functionals of the data-augmentation scheme. Since large values of $1 - r(a)$ result in faster algorithms, it is known as the global speed of EM and is denoted by $s(a)$ (e.g., Meng, 1994).

Our goal is to minimize $r(a)$ or equivalently maximize $s(a)$ as a function of a. Since $I_{\rm obs}$ is independent of the data-augmentation scheme, it is enough to minimize $I_{\rm aug}(a)$ in the sense of a positive semi-definite ordering, as proved in Theorem 2.1.

Theorem 2.1: Suppose $I_{\rm aug}(a) \geq I_{\rm aug}(a')$, that is $I_{\rm aug}(a) - I_{\rm aug}(a')$ is positive semi-definite, then $s(a) \leq s(a')$.

Proof: Since $I_{\rm obs} \geq 0$, $s(a)$ is the smallest eigenvalue of $B(a) \equiv I^{\frac{1}{2}}_{\rm obs}I^{-1}_{\rm aug}(a)I^{\frac{1}{2}}_{\rm obs}$. But $I_{\rm aug}(a) \geq I_{\rm aug}(a')$ implies $B(a) \leq B(a')$ (e.g., Horn and Johnson, 1985, p.470), and thus, the result follows trivially from the Courant-Fischer representation: $s(a) = \min_{b^\top b=1} b^\top B(a) b$. ■

Theorem 2.1 assumes $I_{\text{aug}}(a) - I_{\text{aug}}(a')$ is positive semi-definite, in which case this defines an ordering of the data-augmentation schemes. When $I_{\text{aug}}(a) \geq I_{\text{aug}}(a')$, we may say the augmentation $Y_{\text{aug}}(a')$ is *nested* in $Y_{\text{aug}}(a)$. In such cases, we may write $I - DM^{EM}(a) \equiv S^{EM}(a)$ (the matrix speed of the algorithm) as

$$S^{EM}(a) = I_{\text{obs}} I^{-1}_{\text{aug}}(a') \; I_{\text{aug}}(a') I^{-1}_{\text{aug}}(a) \qquad \textbf{(2.2.1)}$$
$$= S^{EM}(a') \; S^{EM}(a', a),$$

where $S^{EM}(a', a)$ can be viewed as the speed of the EM algorithm with "observed data" $Y_{\text{aug}}(a')$ and augmented data $Y_{\text{aug}}(a)$. (Strictly speaking, this interpretation is not correct because $S^{EM}(a', a)$ is evaluated at $\theta = \theta^\star(Y_{\text{obs}})$, not $\theta = \theta^\star(Y_{\text{aug}}(a'))$, but we will ignore this technical issue which is not important in our search for efficient data-augmentation schemes.) Thus, if the augmentations are nested, not only are the global speeds of convergence appropriately ordered, $s(a) \leq s(a')$, but the matrix speeds of convergence also form the product relationship in (2.2.1). As we shall see in Section 2.4, this is the case with the t-distribution.

Of course, two augmentations need not be nested (i.e., $I_{\text{aug}}(a) - I_{\text{aug}}(a')$ may be neither positive nor negative semi-definite). In such cases $R(a', a) \equiv I_{\text{aug}}(a') I^{-1}_{\text{aug}}(a)$ will be defined as the relative augmented information but does not correspond to the matrix speed of any EM algorithm and (2.2.1) must be rewritten as

$$S^{EM}(a) = S^{EM}(a') R(a', a). \qquad \textbf{(2.2.2)}$$

Intuitively, if $R(a', a)$ is "small", $Y_{\text{aug}}(a')$ results in a faster algorithm than $Y_{\text{aug}}(a)$, and if it is large, the opposite is true. When the augmentations are not

nested, as is the case in the random-effects model described in Section 2.6, Theorem 2.1 does not apply but (2.2.2) may be helpful in selecting an efficient algorithm. In principle, we can directly order the smallest eigenvalues $s(a)$ and do not need to resort to $R(a', a)$ for selection, which does not necessarily provide the correct ordering of $s(a)$. However, it is much easier to deal with $R(a', a)$ because it can be calculated analytically, whereas $s(a)$ is typically intractable analytically. We now turn our attention to two specific examples where these ideas result in algorithms that dramatically reduce the number of iterations required for convergence.

2.3. The t-Model: An Optimal Fitting Algorithm

The multivariate (including univariate) t is a common model for statistical analysis, especially for robust estimation (e.g., Little and Rubin, 1987; Little, 1988; Lange, Little, and Taylor, 1989). Here we let $t_p(\mu, \Sigma, \nu)$ denote a p-dimensional t variable with known degrees of freedom ν and the density

$$f_\nu(x|\mu, \Sigma) \propto |\Sigma|^{-\frac{1}{2}} \left[\nu + (x-\mu)^\top \Sigma^{-1}(x-\mu)\right]^{-\frac{(\nu+p)}{2}}, \quad x \in \mathbb{R}^p.$$

Fitting this model to a data set, $Y_{\text{obs}} = (y_1, \ldots, y_n)$, requires maximizing the likelihood function $\prod_i f_\nu(y_i|\mu, \Sigma)$, which is known to have no general closed-form solution. The EM algorithm provides a simple and stable iterative procedure for carrying out this maximization. The standard implementation of EM relies on the following data-augmentation scheme (see, for example, Dempster, Laird, and Rubin, 1980; Rubin, 1983; Liu and Rubin, 1995b) using the well-known representation of

$t_p(\mu, \Sigma, \nu)$:

$$t_p \equiv t_p(\mu, \Sigma, \nu) = \mu + \frac{\Sigma^{\frac{1}{2}} Z}{\sqrt{q}}, \quad Z \sim N_p(0, I_p), \quad q \sim \chi^2_\nu/\nu, \quad Z \perp q, \qquad (\mathbf{2.3.1})$$

with I_p the p-dimensional identity matrix and " $\perp$ " indicating independence. Now *assume* $y_i, i = 1, \ldots, n$ are i.i.d. realizations of this t_p. Since t_p follows $N_p(\mu, \Sigma/q)$ conditional on q, if we further assume that the $q_i, i = 1, \ldots n$ are observed, that is, $Y_{\text{aug}} = \{(y_i, q_i), i = 1, \ldots, n\}$ is our augmented data, finding the MLE of $\theta \equiv (\mu, \Sigma)$ follows directly from the weighted least-squares procedure given in (2.3.3) and (2.3.4) below. This provides a simple M-step. The E-step finds the expectation of the loglikelihood function of θ based on the augmented data Y_{aug} conditional on Y_{obs} and $\theta^{(t)}$ from the t th [†] iteration of EM. Since this loglikelihood is linear in the "missing" data $Y_{\text{mis}} = (q_1, \ldots, q_n)$, the E-step amounts to calculating

$$w_i^{(t+1)} = \mathrm{E}(q_i | y_i, \mu^{(t)}, \Sigma^{(t)}) = \frac{\nu + p}{\nu + d_i^{(t)}}, \qquad i = 1, \ldots, n, \qquad (\mathbf{2.3.2})$$

where $d_i^{(t)} = (y_i - \mu^{(t)})^\top [\Sigma^{(t)}]^{-1} (y_i - \mu^{(t)})$. Consequently, the standard EM iteration calculates the $(t+1)st$ iterate with

$$\mu^{(t+1)} = \frac{\sum_i w_i^{(t+1)} y_i}{\sum_i w_i^{(t+1)}}, \qquad (\mathbf{2.3.3})$$

$$\Sigma^{(t+1)} = \frac{1}{n} \sum_i w_i^{(t+1)} (y_i - \mu^{(t+1)})(y_i - \mu^{(t+1)})^\top, \qquad (\mathbf{2.3.4})$$

where $w_i^{(t+1)}$ is calculated in (2.3.2). The algorithm then iterates among (2.3.2)–(2.3.4) until it converges.

[†] We continue to use the standard notation of letting t index the iteration. This should not be confused with the t variable or t-model.

Now let us consider a more general data-augmentation scheme by multiplying both the numerator and denominator in (2.3.1) by $|\Sigma|^{-\frac{a}{2}}$, with a being an arbitrary constant, which results in

$$t_p(\mu, \Sigma, \nu) = \mu + \frac{|\Sigma|^{-\frac{a}{2}} \Sigma^{\frac{1}{2}} Z}{\sqrt{q(a)}}, \quad Z \sim N_p(0, I_p), \quad q(a) \sim |\Sigma|^{-a} \chi^2_\nu / \nu, \quad Z \perp q(a). \tag{2.3.5}$$

In other words, we move a portion of the scale factor (this is more transparent for the univariate case, $p = 1$) into the missing data, $q(a)$, where the argument a highlights the fact that its distribution now depends on the *working parameter* a. Note that the standard augmentation scheme (2.3.1) corresponds to $a = 0$ (i.e., $q(0) = q$). Although (2.3.5) is mathematically equivalent to (2.3.1), it provides a different data-augmentation scheme because when $q(a)$ is *assumed* to be known it also contributes to the estimation of Σ. In other words, what (2.3.5) accomplishes is to "transform" part of E-step into the M-step (or vise versa). For each given a, one can proceed as before to derive the corresponding EM algorithm (which may not be easy to implement) and its rate of convergence as a function of a by treating $Y_{\text{aug}}(a) = \{(y_i, q_i(a)), i = 1, \ldots n\}$ as the augmented data. Shortly, we will show that the optimal a that maximizes the speed of the algorithm is $a_{\text{opt}} = 1/(\nu + p)$, a result that is neither obvious nor intuitive (at least to us). Amazingly, the corresponding optimal EM is not only very easy to implement, but in fact only differs from the standard one (2.3.2) - (2.3.4) by a trivial modification, that is, by replacing the denominator n in (2.3.4) with the sum of the weights:

$$\Sigma^{(t+1)}_{\text{opt}} = \frac{\sum_i w_i^{(t+1)} (y_i - \mu^{(t+1)})(y_i - \mu^{(t+1)})^\top}{\sum_i w_i^{(t+1)}}. \tag{2.3.6}$$

This replacement does not change the limit, because $\sum_i w_i^{(t+1)} \to n$ as

$t \to \infty$. This fact is proved by Kent, Tyler and Vardi (1994), who use it to modify one of their EM algorithms for fitting t-distributions. They construct an EM algorithm via a "curious likelihood identity" originally proposed in Kent and Tyler (1991) for transforming a p-dimension location-scale t-distribution into a $(p+1)$-dimensional scale-only t-distribution. They reported that this algorithm converges slower than standard EM (2.3.2) – (2.3.4), but a modification using the aforementioned fact converges faster. We were quite curious about their "curious" and novel construction of that modified EM, and the work presented here provides an answer to such curiosity because their modified EM turns out to be identical to our optimal EM given by (2.3.2), (2.3.3) and (2.3.6). Our derivations not only make it clear that their modified EM is indeed an EM algorithm – and thus possess all the desirable properties of EM (e.g., monotone convergence in likelihood) – but also show why it converges faster than the standard EM for any t-model being fit to any data set, regardless of whether the t-model fits or not. More importantly, the idea of introducing a working parameter seems quite general and (as we will see in Section 2.4) leads to other fast EM algorithms (although its formulation, of course, depends on the particular model being fit).

2.4. The t-Model: Empirical Results and Theory

2.4.1. Simulation Studies

Shortly, we will apply Theorem 2.1 to show theoretically that replacing (2.3.4) with (2.3.6) results in the optimal EM algorithm among algorithms with data-augmentation schemes in the class determined by (2.3.5). Here, by optimal, we mean that it has the fastest asymptotic (with respect to the iteration index, t) global rate of convergence. Such theoretic results provide a general understanding and assurance, but do not tell us how much improvement a user can expect in a typical implementation. (Here, happily, we do not need to consider the extra human effort for implementing the new EM, because there is none.) In addition, since the theoretical rate of convergence of EM only measures the speed of EM near convergence, we have seen instances where examining only the rate of convergence leads to misleading comparisons of the actual number of iterations required for convergence (see Chapter 5 and van Dyk and Meng, 1994).

Therefore, in order to explore the actual gains in computational time, we conducted several simulations. We first generated 100 observations from each of three distributions: (i) $N(0,1)$, (ii) $t_1(0,1,\nu=1)$ (i.e., standard Cauchy), and (iii) a mixture of two thirds $N(0,1)$ and one third exponential with mean 3. We then fit $t_1(\mu,\Sigma,\nu)$ with $\nu=1$ and $\nu=5$ to each data set using both the standard and optimal EM algorithms. Such simulation configurations are intended to reflect the fact that, in reality, there is no guarantee that the data are from a t-model – or even from a symmetric model. (After all, the t-model is often fit in the context

of robust estimation.) We started both algorithms with the same standard initial values, $\mu^{(0)} = \bar{y}$ and $\Sigma^{(0)} = \frac{1}{n}\sum_i (y_i - \bar{y})(y_i - \bar{y})^\top$. (These sample values are well determined, regardless of the underlying model or the model being fit.) We also recorded N_{std} and N_{opt}, the number of iterations required by the standard and optimal algorithms, respectively, for achieving $||\theta^{(t)} - \theta^{(t-1)}||^2 / ||\theta^{(t-1)}||^2 \leq 10^{-10}$, where $\theta = (\mu, \Sigma)$. The simulation was repeated 1000 times and the results appear in Figure 2.1. (Comparing only the number of iterations is often misleading because different algorithms may take more or less time to complete each iteration. In the current case, however, the standard and optimal algorithms clearly require the same amount of computation per iteration.) In all 6000 cases the optimal algorithm was faster than standard EM. Generally the improvement was quite significant. In 5997 cases the improvement was greater than 10% and often reached as high as 50% when the Cauchy model ($\nu = 1$) was fit, the case in which the improvement was most significant. Since EM tends to be slower when ν is smaller in the fitted model, the observed improvement is best when it is most useful.

A second simulation was run to investigate the improvement in higher dimensions. We fit a ten-dimensional Cauchy model to 100 observations generated from $t_{10}(0, V, \nu = 1)$, where V was randomly selected at the outset of the simulation as a positive definite non-diagonal matrix. Using the same starting values and convergence criterion, N_{std} and N_{opt} were again computed for 1000 data sets. Figure 2.2 is a scatter plot of $(N_{\text{std}}, N_{\text{opt}})$ with the improvement $N_{\text{std}}/N_{\text{opt}}$ represented by the dashed lines. The improvement of the optimal EM algorithm is dramatic. Standard EM was at least six-and-a-half times slower in every case

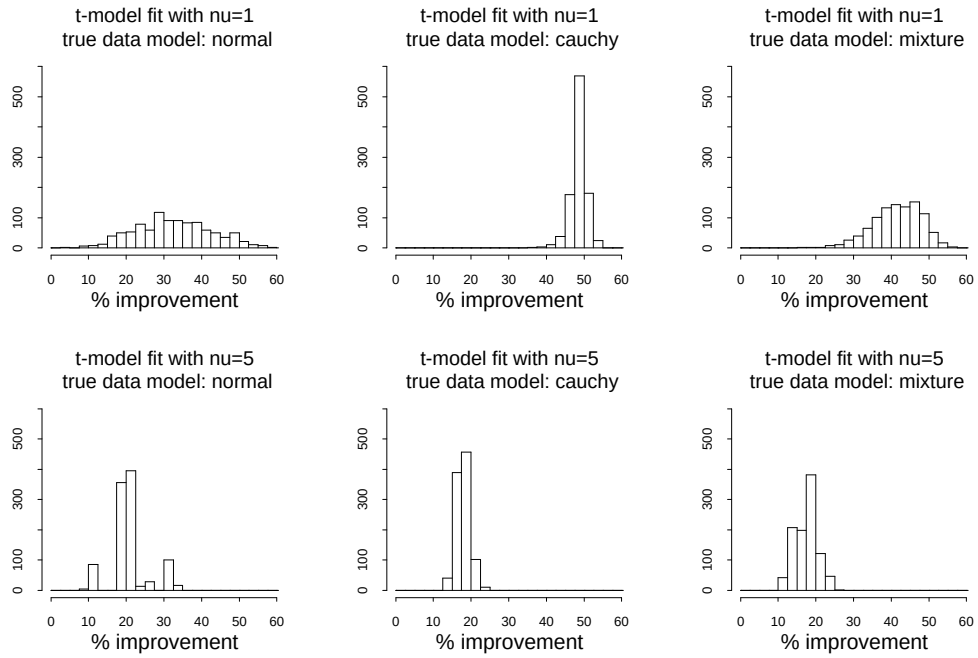

Figure 2.1. The percent improvement of the optimal EM algorithm over the standard EM algorithm for the univariate t-model. Each histogram represents 1000 simulated data sets from one of three models to which one of the two t-distributions was fit with both the standard and optimal algorithms. The histograms show the relative improvement in iterations required for convergence: $\%$ improvement $= 100 \cdot (N_{\text{std}} - N_{\text{opt}})/N_{\text{std}}$.

and was usually between 8 and 10 times slower. Comparing this result with the first simulation, we see that the improvement seems to be much more pronounced in higher dimensional problems. Again, when EM is slowest and improvement is most useful, the gains demonstrated by the optimal algorithm are most striking. It is truly remarkable that such striking gains are obtained without any increase in computation, a true "free lunch"!

One more advantage of the optimal algorithm is worth mentioning. Both algorithms started at the same point, but the optimal algorithm always arrived at $\theta^\star$ in fewer steps. Clearly this is accomplished by taking bigger steps. Figure 2.3

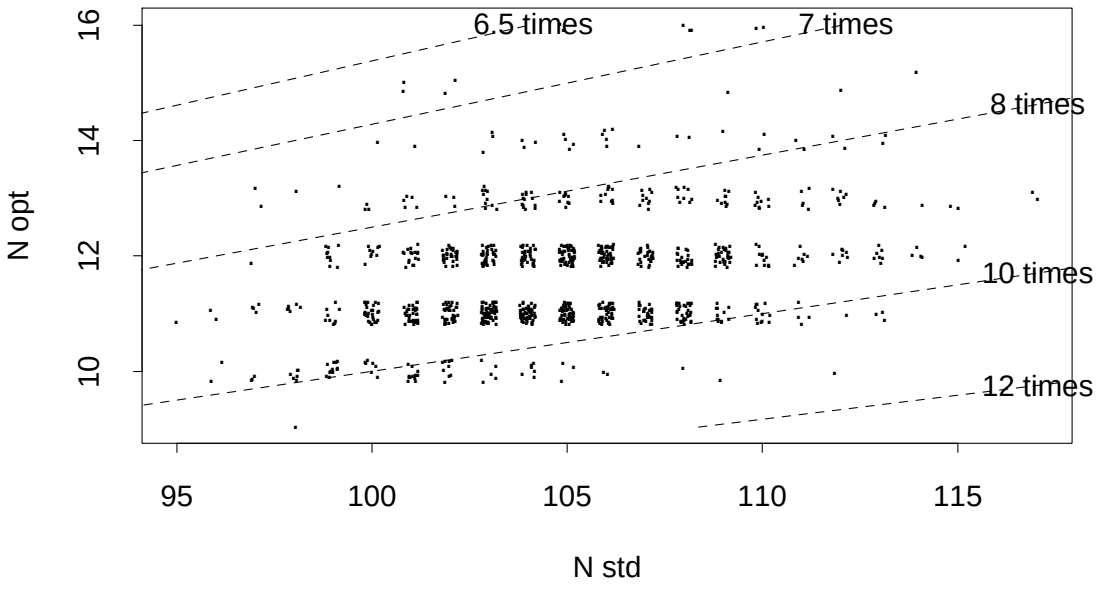

Figure 2.2. The improvement of the optimal EM algorithm over the standard algorithm when fitting $t_{10}(\mu, \Sigma, \nu = 1)$. The plot shows the number of iterations required for the standard EM algorithm N_{std} , and the optimal EM algorithms N_{opt} for each of 1000 simulated 10-variate Cauchy data sets. Because N_{opt} and N_{std} are discrete, the points have been jittered using $Uniform(-0.2, 0.2)$.

illustrates this for one of the Cauchy data sets generated in the univariate simulation. The figure depicts the iterates of the standard and optimal algorithms on the loglikelihood surface and shows how much larger the optimal algorithms steps are, especially during the early iterations. Since we used the step-size convergence criteria, convergence is determined when the step size becomes small and an algorithm that takes big steps is at a disadvantage because convergence will be detected slowly. A close look at the EM iterates reveals that this causes the optimal algorithm to converge to a more precise approximation of $\theta^\star$ than the standard algorithm. That is, the optimal algorithm was closer to $\theta^\star$ than was the standard algorithm when

the convergence criterion was finally satisfied. The optimal algorithm converged not only more quickly but also more precisely.

The difference between the two algorithms stems from the $Q(\theta|\theta^{(t)})$ functions which result from the two augmentation schemes. In particular, the optimal algorithm results from less data augmentation and, hence, a flatter expected augmented-data loglikelihood. This is depicted in Figure 2.4 for the same data set that was used in Figure 2.3 and will be explored analytically in the following section.

2.4.2. Theoretical derivations

It now remains only to show that replacing (2.3.4) with (2.3.6) results in the algorithm that is optimal in the class indexed by a. For a fixed a, the loglikelihood for (μ, Σ) based on the augmented data $Y_{\text{aug}}(a)$ is

$$\begin{aligned} L(\mu, \Sigma|Y_{\text{aug}}(a)) =& \frac{n}{2}[a(p+\nu)-1]\log|\Sigma| \\ & - \frac{|\Sigma|^a \sum_i q_i(a)}{2}\left[\nu + (\bar{y}_w - \mu)^\top \Sigma^{-1}(\bar{y}_w - \mu) + \text{tr}(\Sigma^{-1}S_w)\right], \end{aligned} \tag{2.4.1}$$

where

$$\bar{y}_w = \frac{\sum_i q_i(a) y_i}{\sum_i q_i(a)} \quad \text{and} \quad S_w = \frac{\sum_i q_i(a)(y_i - \bar{y}_w)(y_i - \bar{y}_w)^\top}{\sum_i q_i(a)}. \tag{2.4.2}$$

It follows immediately that the MLE given $Y_{\text{aug}}(a)$ for μ is $\bar{y}_w$. In order to simplify the derivation of the MLE of Σ, we will differentiate (2.4.1) with respect to the elements of $\Psi = \Sigma^{-1}$. The MLE of Σ is the solution of the resulting normal equation

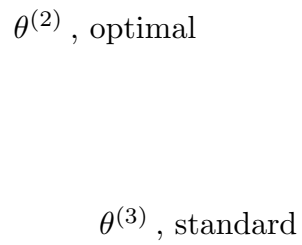

Figure 4. Comparing the optimal and standard iterative mappings. The figure shows the mappings induced on $L(\theta|Y_{\text{obs}})$ by the standard algorithm ($+$) and the optimal algorithm ($\times$) for a one-dimensional Cauchy data set fit with $\nu = 1$, starting from the same $\theta^{(0)}$ (not shown). Notice how much larger the steps are with the optimal algorithm.

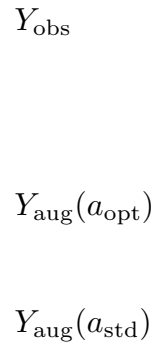

Figure 5. Comparing the loglikelihoods. The plot shows $L(\theta|Y_{\text{obs}})$, as well as $\mathsf{E}[L(\theta|Y_{\text{aug}})|Y_{\text{obs}}, \theta^\star]$ for both the standard and optimal augmentations (each adjusted by their maximum value for comparison). Notice that the optimal augmentation results in a flatter loglikelihood that better approximates $L(\theta|Y_{\text{obs}})$.

$$\frac{\partial}{\partial\Psi}L(\mu,\Sigma|Y_{\text{aug}}(a)) =$$
$$-\frac{n}{2}\left[a(p+\nu)-1\right]\left[2\Sigma-\text{Diag}(\Sigma)\right] \qquad (\textbf{2.4.3})$$
$$-\frac{1}{2}|\Sigma|^a\sum_i q_i(a)[(\bar{y}_w-\mu)(\bar{y}_w-\mu)^\top+2S_w-\text{Diag}(S_w)]$$
$$+\frac{a}{2}\sum_i q_i(a)|\Sigma|^a[2\Sigma-\text{Diag}(\Sigma)][\nu+(\bar{y}_w-\mu)^\top\Sigma^{-1}(\bar{y}_w-\mu)+\text{tr}(\Sigma^{-1}S_w)],$$

which follows from $\frac{\partial}{\partial\Psi}\text{tr}(\Psi S_w)=2S_w-\text{Diag}(S_w)$, and

$$\frac{\partial}{\partial\psi_{ij}}|\Psi| = \begin{cases} 2\Psi_{ij} & \text{if } i\neq j \\ \Psi_{ii} & \text{if } i=j \end{cases}, \qquad (\textbf{2.4.4})$$

where ψ_{ij} is the ij th element of Ψ and Ψ_{ij} is the ij th cofactor of Ψ (Mardia, Kent, and Bibby, 1979, pp. 478-79), and $\text{Diag}(A)$ denotes a diagonal matrix with the same diagonal elements as A. Finally the chain rule along with (2.4.4) and the standard matrix algebra result that $\Sigma=(\Psi_{ij})/|\Psi|$ give us

$$\frac{\partial}{\partial\Psi}|\Psi|^{-a}=a|\Sigma|^a[2\Sigma-\text{Diag}(\Sigma)]$$

and

$$\frac{\partial}{\partial\Psi}\log|\Psi|=2\Sigma-\text{Diag}(\Sigma).$$

Evaluating (2.4.3) at the MLE of $\mu=\bar{y}_w$ and replacing $[2\Sigma-\text{Diag}(\Sigma)]$ with Σ and $[2S_w-\text{Diag}(S_w)]$ with S_w, since we may solve (2.4.3) for each element of Σ individually, we see that the MLE of Σ satisfies

$$\frac{n[a(p+\nu)-1]}{|\Sigma|^a\sum_i q_i(a)}\Sigma+S_w=a[\nu+tr(\Sigma^{-1}S_w)]\Sigma. \qquad (\textbf{2.4.5})$$

Solving (2.4.5) with arbitrary a is quite difficult, but there are two values of a that make (2.4.5) trivial to solve. One is $a=0$, corresponding to the standard

augmentation scheme, which yields $\Sigma = (\sum_i q_i/n)S_w$ and thus (2.3.4). The other is $a_{\text{opt}} = 1/(\nu + p)$, with which the first term on the left side of (2.4.5) is zero, and thus, the solution Σ must be proportional to S_w. It is then easy to verify that the proportionality constant must be one, and therefore the MLE of $\Sigma = S_w$, which yields the corresponding M-step given by (2.3.6). It is arguably a miracle that (2.4.5) can be solved analytically for the optimal a, but for (almost) no other a.

To show that a_{opt} yields the best possible rate of convergence under the augmentation scheme (2.3.5), we apply Theorem 2.1 and need verify only that $I_{\text{aug}}(a) \geq I_{\text{aug}}(a_{\text{opt}})$ for all a. Using techniques similar to those used to derive (2.4.3), we can show

$$I_{\text{aug}}(a) = -\frac{\partial^2}{\partial\theta \cdot \partial\theta} Q(\theta|\theta^\star)\Big|_{\theta=\theta^\star} \tag{2.4.6}$$

$$= \begin{pmatrix} n\Sigma^{\star -1} & 0 \\ 0 & -\frac{n}{2}\left[k_{(i,j;k,l)}\right] \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{n}{2}\vec{\varsigma}\,\vec{\varsigma}^{\top} \end{pmatrix} (a^2(p+\nu) - 2a + 1),$$

where $\theta = (\mu, \text{vec}(\Psi))$ (again $\Psi = \Sigma^{-1}$), the sub-matrix corresponding to vec(Ψ) has rows indexed by i and j and columns indexed by k and l (in reference to the position of a particular parameter in the matrix Ψ), $\vec{\varsigma} = \text{vec}(2\Sigma^\star - \text{Diag}(\Sigma)^\star)$, and

$$k_{(i,j;k,l)} = c_{(i,j;k,l)}(-1)^{(i+j+k+l)}\frac{\Psi_{(i,j;k,l)}}{|\Psi|},$$

with $\Psi_{(i,j;k,l)}$ equal to the determinant of $\Psi^\star$ with distinct rows i and k and distinct columns j and l deleted and $c_{(i,j;k,l)}$ equal to the number of ways that i, j, k and l can be arranged by permuting i and j or k and l so as to result in

the deletion of two distinct rows and two distinct columns. That is,

$$c_{(i,j;k,l)} = \begin{cases} 0 & \text{if three or four of } i,j,k \text{ and } l \text{ are equal} \\ 1 & \text{if there are not three equal but } i=j \text{ and } k=l \\ 2 & \text{if no three are equal, } i\neq j \text{ or } k\neq l, \text{ but there are two equal} \\ 4 & \text{if all four are distinct.} \end{cases}$$

Given (2.4.6) it is easy to show

$$I_{\text{aug}}(a) - I_{\text{aug}}(a_{\text{opt}}) = (\nu+p)\left(a - \frac{1}{\nu+p}\right)^2 C \quad \text{for any } a,$$

where C is the positive semi-definite matrix $\begin{pmatrix} 0 & 0 \\ 0 & \frac{n}{2}\vec{\varsigma}\vec{\varsigma}^{\top} \end{pmatrix}$ which does not depend on a. Thus, the desired result that $a_{\text{opt}} = 1/(\nu+p)$ minimizes the augmented-information is clear. With the note that the E-step for the optimal EM only differs from the standard E-step of (2.3.2) by a scale factor that is independent of i and, thus, is irrelevant for (2.3.3) and (2.3.6), this completes our proof that replacing (2.3.4) by (2.3.6) results in a uniformly faster EM algorithm regardless of ν, p or Y_{obs}.

2.5. Random Effects Models: New Fitting Algorithms

2.5.1. The standard EM algorithm

The random-effects (including variance-component) model is an increasingly common generalization of the standard linear model and is a routine, albeit sometimes notoriously slow, application of the EM algorithm (e.g., Laird and Ware, 1982; Laird et al., 1987; Liu and Rubin, 1995a). Here we assume

$$y_i = X_i^{\top}\beta + Z_i^{\top} b_i + e_i, \quad b_i \sim N_q(0,T), \quad e_i \sim N(0,\sigma^2), \quad b_i \perp e_i, \qquad (\textbf{2.5.1})$$

for $i = 1, \ldots, n$, where X_i $(p \times 1)$ and Z_i $(q \times 1)$ are known covariates; and β are the $(p \times 1)$ fixed effects; $b_i = (b_{i1}, \ldots, b_{iq})$ are the $(q \times 1)$ random effects. Although there is no general closed-from solution for the maximum likelihood estimate $\theta^\star \equiv (\beta^\star, \sigma^{2\star}, T^\star)$ of $\theta \equiv (\beta, \sigma^2, T)$ given $Y_{\text{obs}} = (y_1, \ldots, y_n)$, the EM algorithm again provides a simple and stable fitting algorithm. The standard data augmentation (Dempster, Laird and Rubin, 1977; Laird and Ware, 1982; Laird et al., 1987) which treats the b_i as missing data (i.e., $Y_{\text{aug}} = \{(y_i, b_i); i = 1, \ldots, n\}$) leads naturally to the following algorithm. The E-step calculates the augmented-data sufficient statistics:

$$\hat{b}_i^{(t+1)} = \mathsf{E}(b_i | Y_{\text{obs}}, \theta^{(t)}) = \frac{T^{(t)} Z_i (y_i - X_i^\top \beta^{(t)})}{[\sigma^2]^{(t)} + Z_i^\top T^{(t)} Z_i}, \tag{**2.5.2**}$$

$$\hat{T}_i^{(t+1)} = \mathsf{E}(b_i b_i^\top | Y_{\text{obs}}, \theta^{(t)}) = \hat{b}_i^{(t+1)} \left[\hat{b}_i^{(t+1)}\right]^\top + T^{(t)} - \frac{T^{(t)} Z_i Z_i^\top T^{(t)}}{[\sigma^2]^{(t)} + Z_i^\top T^{(t)} Z_i}. \tag{**2.5.3**}$$

Since $Q(\theta | \theta^{(t)})$ factors into two terms, one involving β and σ^2 and the other involving T, the M-step has a particularly simple form. First we update (β, σ^2) via the linear regression implied by (2.5.1),

$$\beta^{(t+1)} = \left(\sum_{i=1}^n X_i X_i^\top \right)^{-1} \left(\sum_{i=1}^n X_i (y_i - Z_i^\top \hat{b}_i^{(t+1)}) \right) \tag{**2.5.4**}$$

$$\sigma^{2(t+1)} = \frac{1}{n} \sum_{i=1}^n \left[\left(y_i - X_i^\top \beta^{(t+1)} - Z_i^\top \hat{b}_i^{(t+1)} \right)^2 + \operatorname{tr} \left(Z_i Z_i^\top (\hat{T}_i^{(t+1)} - \hat{b}_i^{(t+1)} \left[\hat{b}_i^{(t+1)}\right]^\top) \right) \right].$$

Using the assumed marginal normality of b_i, we then update T with the sums of squares estimate

$$T^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \hat{T}_i^{(t+1)},$$

thus completing a single iteration of the standard algorithm.

2.5.2. A new EM algorithm

In Section 2.3, we rescaled the missing data by a power of its standard deviation, thereby introducing a working parameter into the data augmentation which results in a remarkable EM implementation for the t-model. Inspired by this success, we obviously wanted to try the same idea with the random-effects model given in (2.5.1). Because in this setting the unobserved random variable b can be a vector, the theoretical derivation is much more complicated. In principle, we can rescale b by $T^{-a/2}$, where a is an arbitrary constant, and treat $\{b_i(a) = T^{-a/2}b_i, i = 1, \ldots, n\}$ as the missing data. Since T can be any positive definite matrix and it is difficult to handle an arbitrary power of a matrix, however, the resulting EM algorithm is very difficult if not impossible to implement. This clearly violates our requirement that the resulting algorithm not only needs to be fast but also needs to be simple and stable.

There are two ways of getting around the difficulties that arise from dealing with the matrix scale factor T. The first one is to deal only with $a = 1$. (Recall that the standard algorithm given above corresponds to $a = 0$.) The second is to diagonalize T so as to reduce the problem to q scalar problems. In this section, we will derive the algorithm corresponding to $a = 1$. At first, one might think that this is rather restrictive and may not provide much computational gain. Surprisingly, as we will illustrate in Section 2.6, the simple switch from $a = 0$ to $a = 1$ can dramatically reduce the computation time. Nevertheless, in Section 2.5.3, we will

diagonalize T and create a more flexible class of data-augmentation schemes and, thus, further improve computational efficiency.

To derive the new algorithm, we start by substituting $b_i = S\tilde{b}_i$ into (2.5.1), where S is a symmetric matrix such that $T = S^2$. That is, we express the model as

$$y_i = X_i^\top \beta + Z_i^\top S\tilde{b}_i + e_i, \quad \tilde{b}_i \sim N_q(0, I), \quad e_i \sim N(0, \sigma^2), \quad \tilde{b}_i \perp e_i, \qquad (\mathbf{2.5.5})$$

for $i = 1, \ldots, n$. By doing this, we convert the variance parameter T into the regression parameter S. Note that such a conversion is not one-to-one since S is not unique. However, this does not create a problem for our formulation because we are only using S as an intermediate device for deriving algorithms, and the parameter of interest, namely $\theta = (\beta, \sigma^2, T)$, is uniquely determined and fitted from the model.

Under model (2.5.5) we will treat $Y_{\text{aug}} = \{(y_i, \tilde{b}_i), i = 1, \ldots, n\}$ as the augmented data. Given this data, (2.5.5) can be fit via a simple linear regression with missing values among the predictor variables. To see this more clearly, we re-express the regression model in (2.5.5) as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} X_1^\top \\ X_2^\top \\ \vdots \\ X_n^\top \end{pmatrix} \beta + \begin{pmatrix} \tilde{Z}_1^\top \\ \tilde{Z}_2^\top \\ \vdots \\ \tilde{Z}_n^\top \end{pmatrix} \tilde{S} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}, \qquad (\mathbf{2.5.6})$$

where $\tilde{Z}_i = \text{VEC}(Z_i\tilde{b}_i^\top + \tilde{b}_i Z_i^\top), \tilde{S} = \text{VEC}(S)$, with $\text{VEC}(A)$ being a one-to-one mapping from a $q \times q$ symmetric matrix $A = (a_{ij})$ to a $\tilde{q} = q(q+1)/2$ dimensional vector:

$$\text{VEC}(A) = \left(\frac{a_{11}}{\sqrt{2}}, a_{12}, \ldots, a_{1q}, \frac{a_{22}}{\sqrt{2}}, a_{23}, \ldots, a_{2q}, \ldots, \frac{a_{q-1,q-1}}{\sqrt{2}}, a_{q-1,q}, \frac{a_{q,q}}{\sqrt{2}} \right).$$

Here $\tilde{Z}_i, i = 1, \ldots, n$ are unobserved, but follow $N_{\tilde{q}}(0, \mathsf{Var}(\tilde{Z}_i))$, with $\mathsf{Var}(\tilde{Z}_i)$ completely known because $\tilde{b}_i \sim N_q(0, I)$.

To implement the EM algorithm to fit (2.5.6), with $\tilde{Z} \equiv [\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_n]^\top$ as missing data, we first notice that the augmented-data loglikelihood is linear in $\tilde{Z}$ and $\tilde{B} \equiv \tilde{Z}^\top \tilde{Z} = \sum_{i=1}^{n} \tilde{Z}_i \tilde{Z}_i^\top$. Thus, at the $(t+1)$ st iteration, the E-step computes

$$\begin{aligned} \tilde{Z}_i^{(t+1)} &= \mathsf{E}\left[\tilde{Z}_i | Y_{\text{obs}}, \theta^{(t)}\right] \\ &= \text{VEC}\left(Z_i \mathsf{E}\left[\tilde{b}_i^\top | Y_{\text{obs}}, \theta^{(t)}\right] + \mathsf{E}\left[\tilde{b}_i | Y_{\text{obs}}, \theta^{(t)}\right] Z_i^\top\right) \end{aligned} \qquad \textbf{(2.5.7)}$$

and

$$\tilde{B}_i^{(t+1)} \equiv \mathsf{E}\left[\tilde{Z}_i \tilde{Z}_i^\top | Y_{\text{obs}}, \theta^{(t)}\right], \qquad \textbf{(2.5.8)}$$

for $i = 1, \ldots, n$. The computation of (2.5.7) is straightforward because

$$\begin{aligned} \tilde{b}_i^{(t+1)} \equiv \mathsf{E}\left[\tilde{b}_i | Y_{\text{obs}}, \theta^{(t)}\right] &= \left[S^{(t)}\right]^{-1} \hat{b}_i^{(t+1)} \\ \text{(see (2.5.2))} \qquad &= \frac{S^{(t)} Z_i \left(y_i - X_i^\top \beta^{(t)}\right)}{[\sigma^2]^{(t)} + \left[S^{(t)} Z_i\right]^\top \left[S^{(t)} Z_i\right]}, \end{aligned} \qquad \textbf{(2.5.9)}$$

where $S^{(t)} = \text{VEC}^{-1}(\tilde{S}^{(t)})$. The computation of (2.5.8) is a bit more involved, because $\tilde{B}_i^{(t+1)}$ is not a function of the matrix

$$\begin{aligned} \tilde{T}_i^{(t+1)} &= \mathsf{E}\left[\tilde{b}_i \tilde{b}_i^\top | Y_{\text{obs}}, \theta^{(t)}\right] \qquad \textbf{(2.5.10)} \\ \text{(see (2.5.3))} \quad &= \tilde{b}_i^{(t+1)} \left[\tilde{b}_i^{(t+1)}\right]^\top + I_q - \frac{S^{(t)} Z_i \left[S^{(t)} Z_i\right]^\top}{[\sigma^2]^{(t)} + \left[S^{(t)} Z_i\right]^\top \left[S^{(t)} Z_i\right]}, \end{aligned}$$

but is rather a function of the elements of $\tilde{T}_i^{(t+1)}$, and some "bookkeeping" details are required to express (2.5.8) as a function of the elements of $\tilde{T}_i^{(t+1)}$. In particular, from the definition of the VEC operator, $\tilde{B}_i^{(t+1)}$ is a function of

$$\mathsf{E}\left[(z_{ij}\tilde{b}_{ik} + z_{ik}\tilde{b}_{ij})^2 \middle| Y_{\text{obs}}, \theta^{(t)}\right] = z_{ij}^2 \left[\tilde{T}_i^{(t)}\right]_{kk} + 2 z_{ij} z_{ik} \left[\tilde{T}_i^{(t)}\right]_{jk} + z_{ik}^2 \left[\tilde{T}_i^{(t)}\right]_{jj},$$

where $Z_i = (z_{i1}, \ldots, z_{iq})$, $\tilde{b}_i = (\tilde{b}_{i1}, \ldots, \tilde{b}_{iq})$, and $\left[\tilde{T}_i^{(t)}\right]_{jk}$ is the jk th element of $\tilde{T}_i^{(t)}$.

Once $\tilde{Z}_i^{(t+1)}$ and $\tilde{B}_i^{(t+1)}$ are calculated for $i = 1, \ldots, n$, the M-step finds the maximizer of $Q(\theta|\theta^{(t)})$ as

$$\begin{pmatrix} \beta^{(t+1)} \\ \tilde{S}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} X_i X_i^\top & \sum_{i=1}^{n} X_i \left[\tilde{Z}_i^{(t+1)}\right]^\top \\ \sum_{i=1}^{n} \tilde{Z}_i^{(t+1)} X_i^\top & \sum_{i=1}^{n} \tilde{B}_i^{(t+1)} \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} X_i y_i \\ \sum_{i=1}^{n} \tilde{Z}_i^{(t+1)} y_i \end{pmatrix} \quad (\textbf{2.5.11})$$

and

$$\left[\sigma^2\right]^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \left[\left(y_i - X_i^\top \beta^{(t+1)} - \left[\tilde{Z}_i^{(t+1)}\right]^\top \tilde{S}^{(t+1)} \right)^2 + \operatorname{tr}\left(\tilde{S}^{(t+1)} \left[\tilde{S}^{(t+1)}\right]^\top (\tilde{B}_i^{(t+1)} - \tilde{Z}_i^{(t+1)} \left[\tilde{Z}_i^{(t+1)}\right]^\top) \right) \right]. \quad (\textbf{2.5.12})$$

Computationally, a way to avoid inverting the $(p+\tilde{q}) \times (p+\tilde{q})$ matrix in (2.5.11) is to use the `SWEEP` operator to perform the regression; for details see Little and Rubin (1987, pp. 153-57). Upon convergence, we will compute $T^\star = [S^\star]^2$, which is always positive definite even though $S^\star$ may not be. Furthermore, as we noted earlier, although $S^\star$ is not unique, $T^\star$ is (given the regularity conditions that guarantee the uniqueness of the mode of $L(\theta|Y_{\text{obs}})$).

2.5.3. Implementation of EM after diagonalization

We now describe a second approach, namely we will diagonalize (i.e., orthagonalize) T before we implement the EM algorithm. Let $T = \Delta U \Delta^\top$, where Δ is a lower triangular $(q \times q)$ matrix with ones on the diagonal, and U is a diagonal matrix. It is well-known that such a decomposition exists and is unique

(e.g., Horn and Johnson, 1985, p.162). Let $c_i = \Delta^{-1} b_i$, then $c_i \sim N_q(0, U)$. Since $U \equiv (u_1^2, \ldots, u_q^2)$ is diagonal, we have the flexibility to rescale each element of $c_i = (c_{i1}, \ldots, c_{iq})^\top$ by a power of its own standard deviation. Specifically, for any vector $a = (a_1, \ldots, a_q)^\top \in \mathbb{R}^q$, we can define

$$c_i(a) = \left(\frac{c_{i1}}{u_1^{a_1}}, \frac{c_{i2}}{u_2^{a_2}}, \ldots, \frac{c_{iq}}{u_q^{a_q}} \right)^\top,$$

and treat $Y_{\text{aug}}(a) = \{(y_i, c_i(a)), i = 1, \ldots, n\}$ as the augmented data. For $a = (1, 1, \ldots, 1)$, $c_i(a) = U^{\frac{1}{2}} c_i = \Delta U^{\frac{1}{2}} b_i$. From the representation $T = \Delta U \Delta^\top$, it is clear that this data augmentation stems from using the lower diagonal square root matrix in place of the symmetric square root which was used in the previous section. The re-expression of model (2.5.1) corresponding to this data augmentation is

$$y_i = X_i^\top \beta + \sum_{j=1}^{q} \sum_{k=j}^{q} c_{ij}(a) z_{ik} \delta_{kj} u_j^{a_j} + e_i, \qquad \textbf{(2.5.13)}$$

where $\Delta = (\delta_{kj})$ and $c_i(a) \equiv (c_{i1}(a), \ldots, c_{iq}(a))^\top$. Although we can in principle implement the EM algorithm for any $a \in \mathbb{R}^q$ which will result in a fast algorithm, we will focus on $a \in \{0, 1\}^q$. That is, a_i can only take on values 0 or 1 in order to keep the resulting algorithms simple to implement which is one of the main objectives of our search. Within this class of data-augmentation schemes, given $Y_{\text{aug}}(a)$, (2.5.13) is a linear regression with $p + \dfrac{q(q-1)}{2} + \sum_{j=1}^{q} a_j$ regression coefficients when we view $\{\delta_{kj} u_j, k \geq j, \text{ for } a_j = 1\} \cup \{\delta_{kj}; k > j, \text{ for } a_j = 0\}$ as the $\dfrac{q(q-1)}{2} + \sum_{j=1}^{q} a_j$ regression coefficients besides β. The M-step thus has two parts. First, we update $(\beta, \sigma^2, \Delta, \{u_j, \text{ for } a_j = 1\})$ via the linear regression (2.5.13), performed in the same way as described in Section 2.5.2, by treating $\{c_{ij}(a) z_{ik}, k > j\} \cup \{c_{ij}(a) z_{ij}, \text{ for } a_j =$

1} as the missing covariates. For example, for $a = (1, 1, \ldots, 1)$, we can rewrite (2.5.13) as

$$y_i = X_i^\top \beta + \tilde{X}_i^\top \tilde{\beta} + e_i,$$

where $\tilde{X}_i$ is a vector with components $c_{ij}(a) z_{ik}$ for $j = 1, \ldots, q$, $k \geq j$ and $\tilde{\beta}$ is a vector with corresponding components $\delta_{kj} u_j$. In this case, the parameters can be updated by

$$\begin{pmatrix} \beta^{(t+1)} \\ \tilde{\beta}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n X_i X_i^\top & \sum_{i=1}^n X_i \left[\tilde{X}_i^{(t+1)}\right]^\top \\ \sum_{i=1}^n \tilde{X}_i^{(t+1)} X_i^\top & \sum_{i=1}^n \tilde{B}_i^{(t+1)} \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n X_i y_i \\ \sum_{i=1}^n \tilde{X}_i^{(t+1)} y_i \end{pmatrix} \quad (\mathbf{2.5.14})$$

and

$$\begin{aligned} \left[\sigma^2\right]^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \Bigg[& \left(y_i - X_i^\top \beta^{(t+1)} - \left[\tilde{X}_i^{(t+1)}\right]^\top \tilde{\beta}^{(t+1)} \right)^2 \\ & + \operatorname{tr} \left(\tilde{\beta}^{(t+1)} \left[\tilde{\beta}^{(t+1)}\right]^\top (\tilde{B}_i^{(t+1)} - \tilde{X}_i^{(t+1)} \left[\tilde{X}_i^{(t+1)}\right]^\top) \right) \Bigg], \quad (\mathbf{2.5.15}) \end{aligned}$$

where $\tilde{X}_i^{(t+1)} = \mathsf{E}\left[\tilde{X}_i | Y_{\text{obs}}, \theta^{(t)}\right]$ and $\tilde{B}_i^{(t+1)} = \mathsf{E}\left[\tilde{X}_i \tilde{X}_i^\top | Y_{\text{obs}}, \theta^{(t)}\right]$. Second, (for any $a \in \{0,1\}^q$) we update $\{u_j, \text{ for } a_j = 0\}$ by using $c_{ij}(a) \sim N(0, u_j^2)$ when $a_j = 0$ and thus

$$[u_j^{(t+1)}]^2 = \frac{1}{n} \sum_{i=1}^n \mathsf{E}\left[c_{ij}^2(a) | Y_{\text{obs}}, \theta^{(t)}\right], \quad \text{for } j \text{ such that } a_j = 0. \quad (\mathbf{2.5.16})$$

The E-step is also quite similar to that described in Section 2.5.1. First we calculate (corresponding to (2.5.9)–(2.5.11) or (2.5.2)–(2.5.3))

$$\hat{c}_i^{(t+1)}(a) = \mathsf{E}\left[c_i(a) | Y_{\text{obs}}, \theta^{(t)}\right] = \frac{\tilde{U}^{(t)}(2-a)\left[\Delta^{(t)}\right]^\top \left(y_i - X_i^\top \beta^{(t)}\right)}{[\sigma^2]^{(t)} + Z_i^\top \Delta^{(t)} U^{(t)} [\Delta^{(t)}]^\top Z_i} \quad (\mathbf{2.5.17})$$

and

$$\begin{aligned}\hat{U}_i^{(t+1)}(a) &= \mathsf{E}\left[c_i(a)c_i^\top(a)|Y_{\text{obs}},\theta^{(t)}\right] \\ &= \hat{c}_i^{(t+1)}(a)\left[\hat{c}_i^{(t+1)}(a)\right]^\top + \tilde{U}^{(t)}(2(1-a)) \\ &\quad - \frac{\tilde{U}^{(t)}(2-a)\Delta^{(t)}Z_i\left[\tilde{U}^{(t)}(2-a)\Delta^{(t)}Z_i\right]^\top}{[\sigma^2]^{(t)} + Z_i^\top\Delta^{(t)}U^{(t)}\left[\Delta^{(t)}\right]Z_i},\end{aligned} \tag{2.5.18}$$

where $\tilde{U}(d) \equiv \text{Diag}\left\{\left[u_1^{(t)}\right]^{d_1},\ldots,\left[u_q^{(t)}\right]^{d_q}\right\}$ for $d=(d_1,\ldots,d_q)$. We then use the elements of $\hat{c}_i^{(t+1)}(a)$ and $\hat{U}_i^{(t+1)}(a)$, $i=1,\ldots,n$ to calculate the augmented-data sufficient statistics. In particular, $\mathsf{E}\left[c_{ij}^2(a)|Y_{\text{obs}},\theta^{(t)}\right]$ needed for (2.5.16) is simply the j th diagonal term of $\hat{U}_i^{(t+1)}(a)$, and the augmented-data sufficient statistics needed for the regression (i.e., the input for the `SWEEP` operator or the terms needed in (2.5.14) and (2.5.15)) are

$$\mathsf{E}\left[c_{ij}(a)z_{ik}\big|Y_{\text{obs}},\theta^{(t)}\right] = z_{ik}\hat{c}_{ij}^{(t+1)}(a),$$

for $j=1,\ldots,q$ and $k\geq j$ and

$$\mathsf{E}\left[c_{ij}(a)z_{ik}c_{il}(a)z_{im}\big|Y_{\text{obs}},\theta^{(t)}\right] = z_{ik}z_{im}\left[\hat{U}_i^{(t+1)}(a)\right]_{jl},$$

for $j=1,\ldots,q$, $k\geq j$, $l=1,\ldots,q$ and $m\geq l$, where $\hat{c}_{ij}^{(t+1)}(a)$ is the j th component of the vector $\hat{c}_i^{(t+1)}(a)$ and $\left[\hat{U}_i^{(t+1)}(a)\right]_{jl}$ is the jl th element of $\hat{U}_i^{(t+1)}(a)$.

Once the algorithm has converged, it is easy to compute the original parameter $T=\mathsf{Var}(b)=\mathsf{Var}(\Delta c)=\Delta U\Delta^\top$. It should be noted that fitting the regression model (2.5.13) can result in negative values for the $\{u_j^\star, j=1,\ldots,q\}$. This should not be cause for alarm, however, since $\Delta^\star U^\star\Delta^{*\top}$ will remain positive definite and unique as long as $T^\star$ is. In fact, since Δ and U are unique for each T, there are

exactly 2^q modes of $L(\beta, \sigma^2, U, \Delta|Y_{\text{obs}})$ (corresponding to the 2^q diagonal roots of U) for every mode of $L(\beta, \sigma^2, T|Y_{\text{obs}})$.

We now have $2^q + 2$ algorithms that are straightforward to implement and which will generally converge to a local maximum of $L(\theta|Y_{\text{obs}})$. In order to evaluate the relative computational merits of the algorithms, we will first present an empirical study and then analyze the algorithms in terms of their matrix and global rates of convergence.

2.6. Random Effects Models: Empirical Results and Theory

2.6.1. Simulation Studies

Two sets of empirical studies were conducted, each with data generated from the model

$$y_i = x_{i1}\beta_1 + x_{i2}\beta_2 + z_{i1}b_{i1} + z_{i2}b_{i2} + e_i, \qquad (\textbf{2.6.1})$$

where $x_{i1} = 1$, $x_{i2} = i$, $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \sim N_2\left(0, \begin{pmatrix} 4 & 0 \\ 0 & 9 \end{pmatrix}\right)$, and $e_i \sim N(0, \sigma^2)$, with b_i and e_i independent. In the first set of studies, (2.6.1) was treated as a variance-component model (with covariates) and (z_{i1}, z_{i2}) took the four values in $\{0, 1\}^2$ in equal proportion. In the second set of simulations, (2.6.1) was treated as a random-effects model and the z_{ij} were generated independently from a standard normal distribution at each replication.

As we shall see, the relative efficiency of the algorithms depends on the rela-

tive sizes of the random effects and the residual variance. The variance-component study was therefore repeated with $\sigma^2 = 1,\ 4,\ 9$ and 36. For each of these values, we generated 100 observations from (2.6.1). The starting values $\beta^{(0)}$ and $\left[\sigma^2\right]^{(0)}$ were obtained by fitting (2.6.1), ignoring the variance components, and $T^{(0)}$ was set to $\begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}$. We ran the standard algorithm along with EM $_{(0,0)}$ and EM $_{(1,1)}$, that is the new algorithm with $Y_{\text{aug}}((0,0))$ and $Y_{\text{aug}}((1,1))$ respectively (i.e., with $a = (0,0)$ and $a = (1,1)$), and recorded N_{std}, $N_{(0,0)}$, and $N_{(1,1)}$ – the number of iterations required by each algorithm before the convergence criterion $L(\theta^{(t)}|Y_{\text{obs}}) - L(\theta^{(t-1)}|Y_{\text{obs}}) < 10^{-7}$ was reached. The simulation was repeated 200 times and the results appear in Figure 2.5.

The scatter plots in the first column of the figure compares $N_{(0,0)}$ with N_{std}. When $\sigma^2 = 1$, using $Y_{\text{aug}}((0,0))$ requires slightly more iteration than the standard algorithm, but the two algorithms seem quite comparable. The second column of Figure 2.5 compares $N_{(1,1)}$ with N_{std} and highlights the great computational savings EM $_{(1,1)}$ offers over the standard algorithm, especially when σ^2 is large relative to T. In particular, with $\sigma^2 = 36$ (about 5.5 times the average random effect, $\frac{1}{n}\sum_i Z_i^\top T Z_i$), it was not unusual for the standard algorithm to require 100 times more iterations to converge, and sometimes it took 600 times more. Since all of the algorithms require roughly the same computational time per iteration, this translates into real computational savings. For example, an older Sun Workstation might compute about 30 iterations of any one of the algorithms per second, in which case the data cloud to the right of the scatter plot corresponding to EM $_{(1,1)}$ with $\sigma^2 = 36$ represents 12.5 minutes being cut to between 1 and 10 seconds.

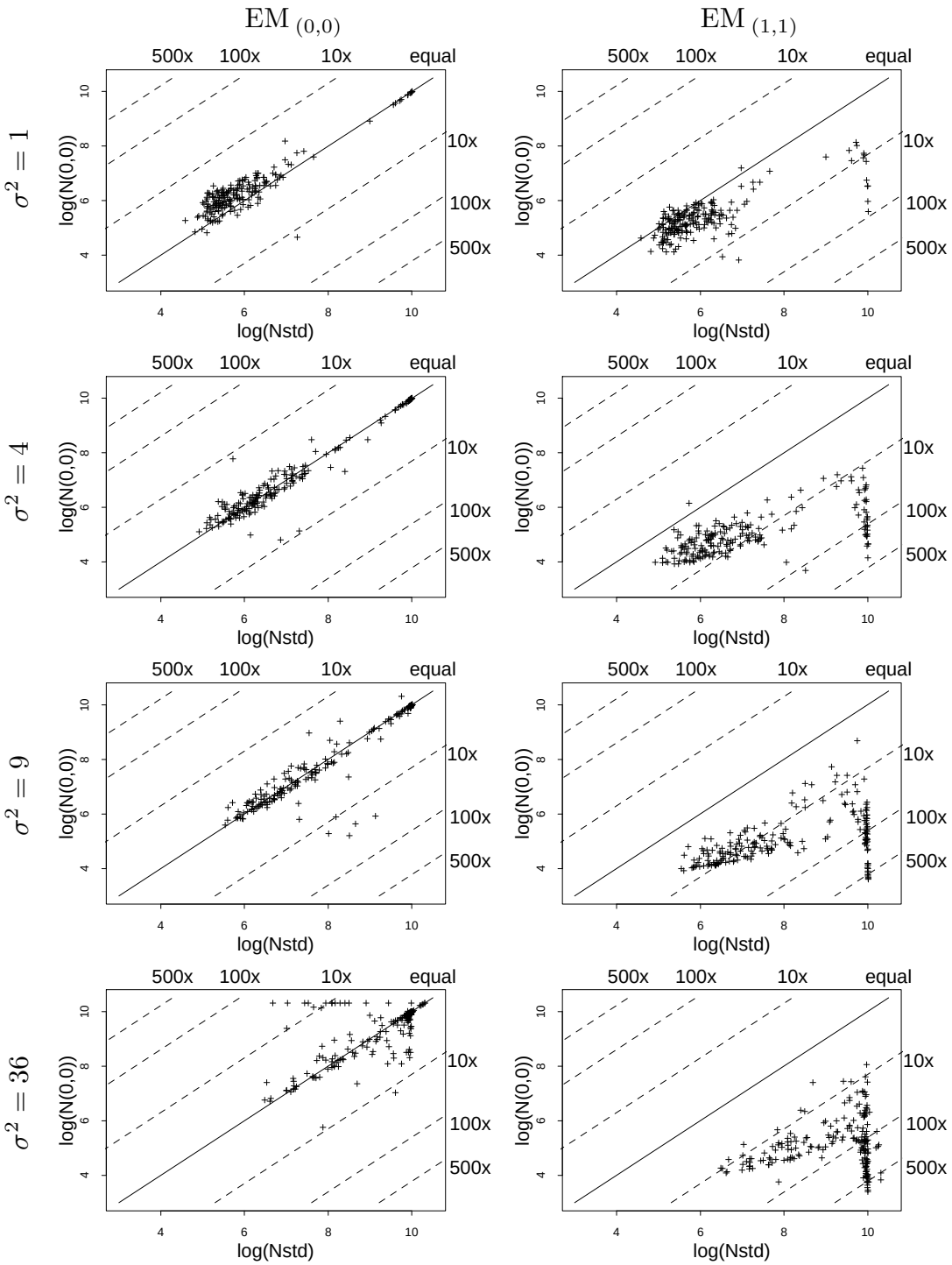

Figure 2.5. Comparing the log number of iterations required by EM $_{(0,0)}$ and EM $_{(1,1)}$ for convergence with that of the standard algorithm. In the plot "100x" means "100 times", etc. On a computer that runs 30 iterations per second, the tick marks 4, 6, 8, and 10 on the X and Y axes correspond to 1.8 seconds, 13.4 seconds, 1.7 minutes, and 12.2 minutes.

Even when $\sigma^2 = 1$ (less than one sixth the average random effect) EM $_{(1,1)}$ tended to be slightly more efficient than the standard algorithm. When the residual variance is small, however, using EM $_{(1,1)}$ tends to offer more modest gains over the standard algorithm. In order to examine how the algorithms compare when the residual variance is very small, the simulation was repeated with $\sigma^2 = 1/9$ (about one sixtieth of the average random effect). The results appear in Figure 2.6 and indicate that the standard algorithm tends to be about 2.7 times faster than EM $_{(1,1)}$. Note, however, that $N_{(1,1)}$ is centered around $e^7 \approx 1100$ iterations (37 seconds at 30 iterations per second) compared with the $e^{10} \approx 22000$ (12.2 minutes) when the standard algorithm was slow (see Figure 2.5). In this simulation, the standard algorithm seems to be more efficient only when both algorithms are relatively fast. Whereas EM $_{(1,1)}$ is more efficient when the standard algorithm is very slow and improvement is badly needed (see Figure 2.7). Thus, unless the residual variance is very small relative to the random effects, EM $_{(1,1)}$ seems to be the algorithm to choose.

The second set of simulations was identical to the first except that in order to look at the more general random-effects problem, the z_{ij} were independently generated from a standard normal distribution. The simulation was repeated for $\sigma^2 =$ 0.25, 1, 4, 9, 16, 25, 36, 49, 64, and 81. Again, we want to compare the efficiency of the standard EM algorithm with EM $_{(1,1)}$ as a function of the relative sizes of the residual variance and the random effects. In the previous simulation T and $\{Z_i, i = 1, \ldots, n\}$ were fixed, and it sufficed to consider efficiency as a function of σ^2. Since $\{Z_i, i = 1, \ldots, n\}$ changes at each replication in the current simula-

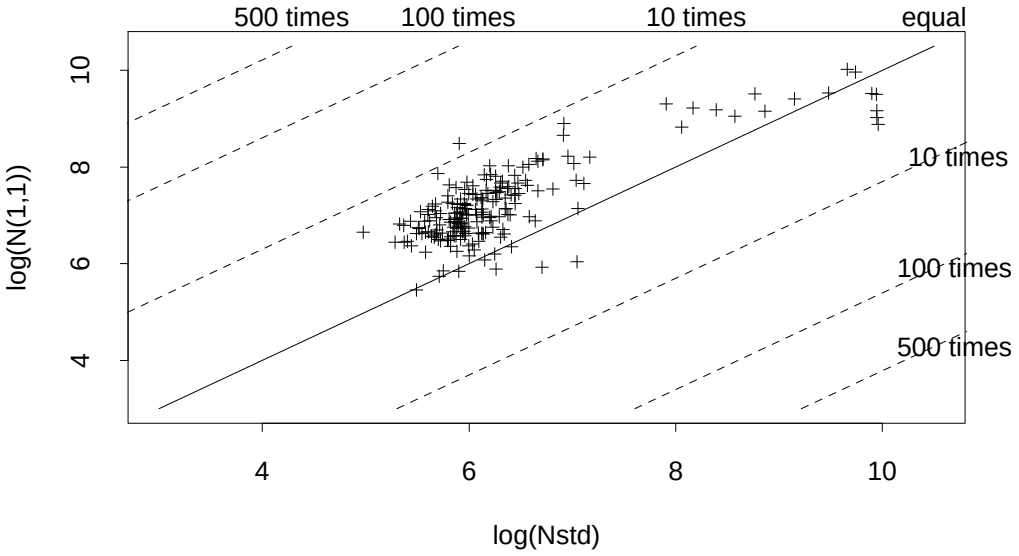

Figure 2.6. Performance of EM $_{(1,1)}$ when the residual variance is very small. Here $\sigma^2 = 1/9$ and the average random effect is 6.5. Note that the median number of iterations required by EM $_{(1,1)}$ is $e^7 = 1142$ (38 seconds) which is less efficient than the standard algorithms $e^{5.26} = 194$ iterations (6.4 seconds).

tion, we look at the log of the number of iterations required for convergence as a function of $\log(\sigma^{2\star}/\frac{1}{n}\sum_i Z_i^\top T^\star Z_i)$. (This expression is evaluated at the MLE since the matrix rate of convergence, $I - I_{\rm obs} I_{\rm aug}^{-1}$ is evaluated at the MLE.) Figure 2.8 displays a sequence of scatter plots. The first displays the efficiency of the standard algorithm, which does well only when the residual variance is somewhat smaller than the average random effect. The second scatter plot looks at EM $_{(1,1)}$ which continues to do very well when the residual variance is large, but does not perform as well when the residual variance is small. Note that when the residual variance was small, both algorithms perform poorly in this simulation, as opposed to the variance-component simulation in which both algorithms performed well. The final

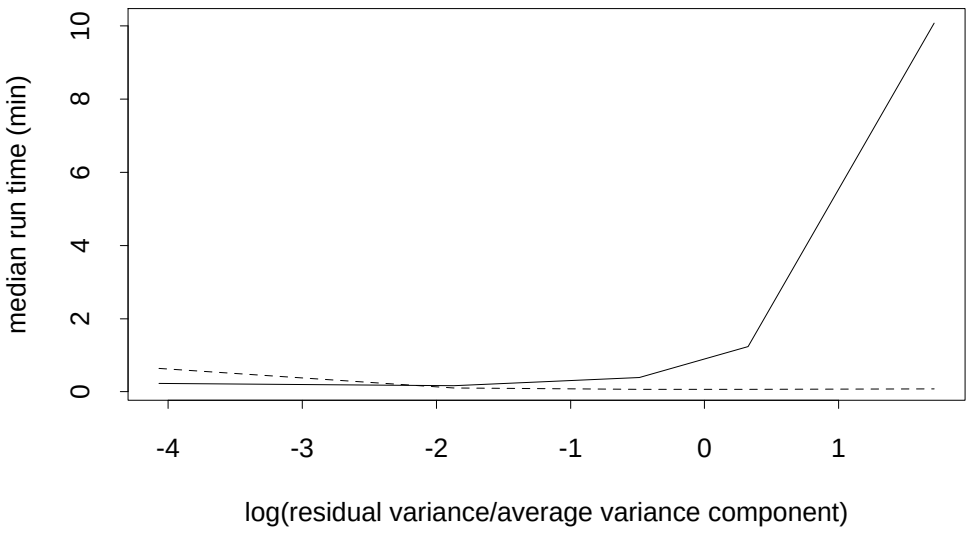

Figure 2.7. The median run time at 30 iterations per second. The dashed line represents EM $_{(1,1)}$ and the solid line represents the standard algorithm. When the residual variance is very small the standard algorithm is somewhat more efficient, but the gain is trivial relative to the the improvement of EM $_{(1,1)}$ when the residual variance is moderate to large.

plot in Figure 2.8 compares the two algorithms. When the the residual variance is less than about one-tenth of the average random effects, the standard algorithm tends to slightly outperform EM $_{(1,1)}$ (as much as 3.4 times faster). On the other hand, when the random effects do not dominate the residual variance, EM $_{(1,1)}$ is clearly superior (as much as 1034 times faster).

Although the relative gain of the standard algorithm over EM $_{(1,1)}$ is small even when the residual variance is very small, cutting the computational time even in half can be significant since both algorithms are so slow in this case. In order to take advantage of the standard algorithm when it is more efficient, a preliminary approximation of $\theta^\star$ can be used to decide between the two algorithms. In order

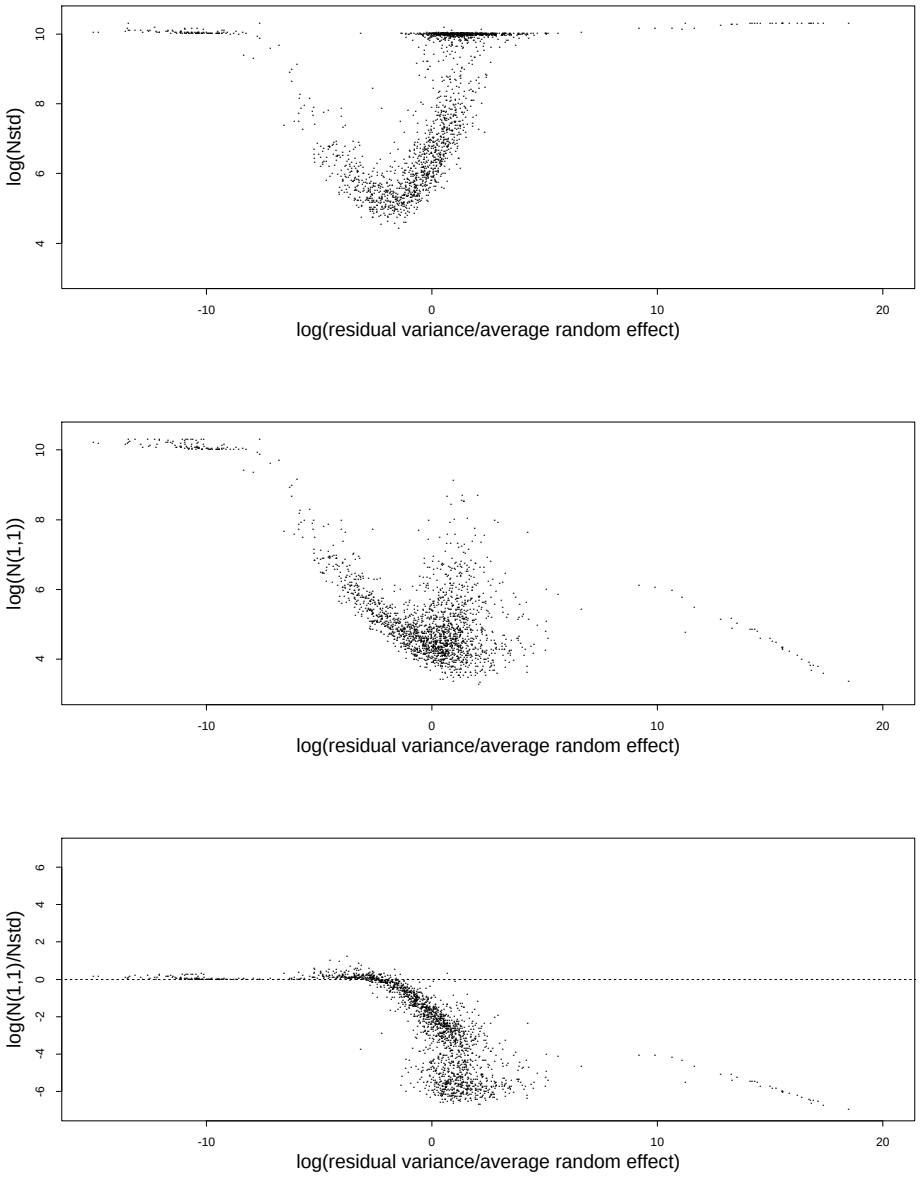

Figure 2.8. Iterations required by EM as a function of the log of the fitted residual variance relative to the average fitted random effect (i.e. $\frac{1}{n}\sum_i Z_i^\top T^\star Z_i$). The first plot reports the log of the number of iterations required by the standard algorithm, the second the log of the number of iterations required by EM $_{(1,1)}$, and the third relative number. The standard algorithm performs better when the residual variance is small, and EM $_{(1,1)}$ performs better when the random effects are small. On a computer that runs 30 iterations per second, the tick marks 4, 6, 8, and 10 on the Y axis in the first two plots correspond to 1.8 seconds, 13.4 seconds, 1.7 minutes, and 12.2 minutes.

to investigate this, we repeated the random-effects simulation (with new random seeds) with an adaptive algorithm, which first runs $\text{EM}_{(1,1)}$ for 20 iterations and then switches to the standard algorithm if

$$4\left[\sigma^2\right]^{(20)} \leq \frac{1}{n}\sum_i Z_i^\top T^{(20)} Z_i. \qquad (\mathbf{2.6.2})$$

This criterion is based on the result of the following section that when T is assumed diagonal in the fitted model, a_j should be set to zero in the data-augmentation scheme (2.5.13) only if $2\sigma^{2\star} \leq \frac{\tau_j^{2\star}}{n}\sum_{i=1}^n z_{ij}^2$, where $\tau_j^{2\star}$ is the MLE of the jth diagonal term of T. If we add this expression over $j = 1, \ldots, q$ and adjust for non-diagonal T, we obtain (2.6.2) with the 4 being replaced by $2q$. As Figure 2.9 indicates, this algorithm almost always switched to the standard algorithm when it was beneficial to do so. (Surprisingly, the switched algorithm was often slightly faster than the pure standard algorithm.) Since this adaptive algorithm is easy to implement and generally performs well against both $\text{EM}_{(1,1)}$ and the standard algorithm, we recommend its use in practice.

Finally, we compare the algorithm of Section 2.5 to both the standard algorithm and $\text{EM}_{(1,1)}$, using the simulation with randomly generated Z_i which was described earlier. Figure 2.10 shows that in terms of the number of iterations required for convergence this algorithm performs as well or better than $\text{EM}_{(1,1)}$. The final plot in Figure 2.10, however, reveals that there is a problem with this algorithm. Although the number of iterations required for convergence is comparable to $\text{EM}_{(1,1)}$, each iteration is more expensive. This stems from the more complicated bookkeeping required in implementing the M-step. Reduced overall computation time along with the added versatility of being able to rescale some

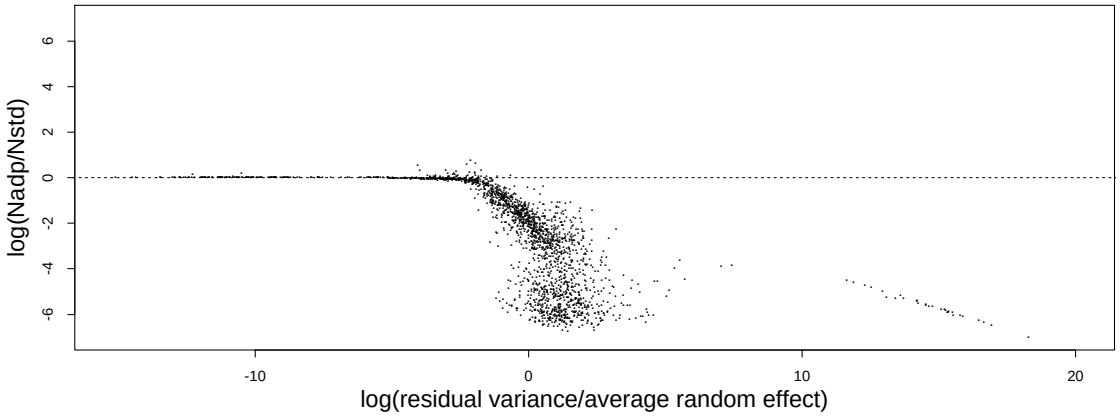

Figure 2.9. Picking the fastest algorithm. After 20 iterations of EM $_{(1,1)}$, the current approximation $\theta^{(20)}$ was used to determine which algorithm should be used. If $4\left[\sigma^2\right]^{(20)} < \frac{1}{n}\sum_i Z_i^\top T^{(20)} Z_i$, the standard algorithm was used until convergence. Otherwise, we continued with EM $_{(1,1)}$ until convergence. Notice that this procedure almost always resulted in an algorithm faster that the standard algorithm.

(but not necessarily all) of the random effects makes EM $_{(1,1)}$ more attractive than this algorithm.

2.6.2. Theoretical derivations

The theory behind choosing an efficient augmentation scheme for the random-effects model fit with the algorithms described in Section 2.5.2 is considerably more complicated then for the t-models presented in Section 2.4.2. The main difficulty is that the expected augmented-data information matrix, $I_{\text{aug}}(a)$ is generally of large dimension and has a complicated structure. Specifically, the dimension of the parameter $\theta = (\beta, \Delta, U, \sigma^2)$ is $p + \dfrac{q(q+1)}{2} + 1$, and $I_{\text{aug}}(a)$ consists of the following

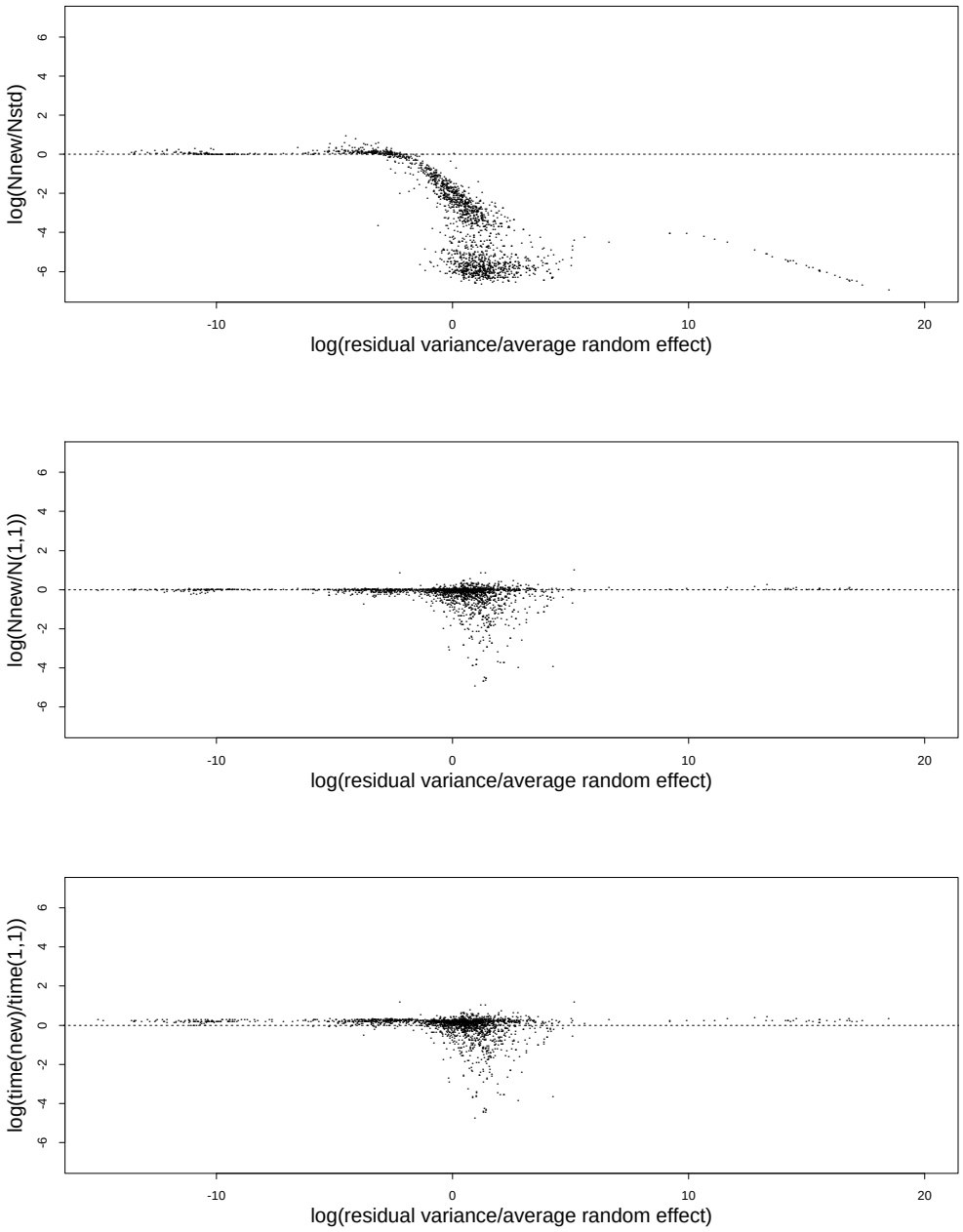

Figure 2.10. Scaling the random effects without diagonalizing T. The first two plots compare the number of iterations required for convergence by the algorithm presented in Section 2.5.2 with the standard algorithm and EM $_{(1,1)}$ and show that this algorithm tends to perform as well, or somewhat better than, EM $_{(1,1)}$. The final plot, however, compares the computational time required by this algorithm with that of EM $_{(1,1)}$ and shows that because the cost per iteration is higher, this algorithm does not generally perform as well as EM $_{(1,1)}$.

submatrices

$$I_{\text{aug}}(a) = \begin{pmatrix} I_{\beta\beta}(a) & I_{\beta\Delta}(a) & I_{\beta U}(a) & I_{\beta\sigma^2}(a) \\ I_{\beta\Delta}^\top(a) & I_{\Delta\Delta}(a) & I_{\Delta U}(a) & I_{\Delta\sigma^2}(a) \\ I_{\beta U}^\top(a) & I_{\Delta U}^\top(a) & I_{UU}(a) & I_{U\sigma^2}(a) \\ I_{\beta\sigma^2}^\top(a) & I_{\Delta\sigma^2}^\top(a) & I_{U\sigma^2}^\top(a) & I_{\sigma^2\sigma^2}(a) \end{pmatrix}.$$

It is not difficult to show, by differentiating the expected augmented-data loglikelihood

$$\begin{aligned} Q(\beta, \Delta, U, \sigma^2 | \theta^{(t)}) = & -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \mathsf{E}\left[\left(y_i - X_i^\top \beta - Z_i^\top \Delta \tilde{U}(a) c_i(a) \right)^2 \Big| \theta^{(t)}, Y_{\text{obs}} \right] \\ & - \frac{n}{2} \log(\sigma^2) - \frac{n}{2} \sum_{j=1}^{q} (1 - a_j) \log(u_j^2) \qquad \textbf{(2.6.3)} \\ & - \frac{1}{2} \sum_{i=1}^{n} \mathsf{E}\left[c_i(a)^\top \tilde{U}^{-1} (2(1-a)) c_i(a) \big| \theta^{(t)}, Y_{\text{obs}} \right], \end{aligned}$$

where $\tilde{U}(a) = \text{Diag}\left\{u_1^{a_1}, \ldots, u_q^{a_q}\right\}$ for $a = (a_1, \ldots, a_q)$ (thus $U = \tilde{U}(2, 2, \ldots, 2)$), that we have (when evaluated at $\theta = \theta^\star$) $I_{\beta\sigma^2}(a) = 0$, $I_{\Delta\sigma^2}(a) = 0$, $I_{U\sigma^2}(a) = 0$, and $I_{\beta\beta}(a)$ and $I_{\sigma^2\sigma^2}$ do not depend on a. Furthermore, when $\mathsf{E}(y_i | X_i, Z_i) = X_i^\top \beta$, that is, when the mean structure of the posited model is correct for the data, $\lim_{n\to\infty} \frac{1}{n} I_{\beta\Delta}(a) = 0$ and $\lim_{n\to\infty} \frac{1}{n} I_{\beta U}(a) = 0$. Thus, as long as n is not too small, the only part of $I_{\text{aug}}(a)$ that can change substantially with a is the $\frac{q(q+1)}{2} \times \frac{q(q+1)}{2}$ submatrix

$$\widetilde{I_{\text{aug}}}(a) = \begin{pmatrix} I_{\Delta\Delta}(a) & I_{\Delta U}(a) \\ I_{\Delta U}^\top(a) & I_{UU}(a) \end{pmatrix}. \qquad \textbf{(2.6.4)}$$

In fact, even when $I_{\beta\Delta}(a)$ or $I_{\beta U}(a)$ are nonzero, we expect they have little impact on the smallest eigenvalue of the speed matrix relative to the impact of $\widetilde{I_{\text{aug}}}(a)$ because the positiveness of $I_{\text{aug}}(a)$ requires that any off-diagonal blocks be dominated by the diagonal blocks. We will thus focus on (2.6.4) when we search for optimal, or good, values of a.

2.6.3. Theoretical derivaton with one random effect

We will attempt to apply Theorem 2.1 which requires us to order $I_{\text{aug}}(a)$, which, as we have seen, is (approximately) equivalent to ordering $\widetilde{I_{\text{aug}}}(a)$. We start with the simple case of one random effect (i.e., $q = 1$), in which case $\widetilde{I_{\text{aug}}}(a)$ is a scalar and (2.6.3) reduces to

$$
\begin{aligned}
Q(\beta, u^2, \sigma^2 | \theta^{(t)}) = \qquad & \qquad (\mathbf{2.6.5}) \\
& - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left[\left(y_i - X_i^\top \beta\right)^2 - 2 \left(y_i - X_i^\top \beta\right) z_i u^a [\hat{c}_i^{(t+1)}(a)] + z_i^2 u^{2a} [\hat{u}_i^{(t+1)}(a)] \right] \\
& - \frac{n}{2} \log(\sigma^2) - \frac{n}{2}(1-a) \log(u^2) - \frac{1}{2} \sum_{i=1}^{n} \frac{\hat{u}_i^{(t+1)}(a)}{u^{2(1-a)}},
\end{aligned}
$$

where $\hat{u}_i^{(t+1)}(a)$ is the scalar version of (2.5.18). In order to derive an expression for $\widetilde{I_{\text{aug}}}(a)$, we differentiate (2.6.5) twice with respect to u^2:

$$
\begin{aligned}
& -\frac{\partial^2 Q}{(\partial u^2)^2} \\
& = \frac{1}{4\sigma^2 u^4} \sum_{i=1}^{n} \left[-(y_i - X_i^\top \beta) z_i [\hat{c}_i^{(t+1)}(a)] u^a (a^2 - 2a) + 2 z_i^2 [\hat{u}_i^{(t+1)}(a)] u^{2a} (a^2 - a) \right] \\
& - \frac{n}{2u^4} \left[(1-a) - \frac{1}{n} \sum_{i=1}^{n} \frac{[\hat{u}_i^{(t+1)}(a)]}{u^{2-2a}} (a-1)(a-2) \right]. \qquad (\mathbf{2.6.6})
\end{aligned}
$$

$\widetilde{I_{\rm aug}}(a)$ is equal to (2.6.6) evaluated at $\theta = \theta^\star$, in which case

$$\hat{c}_i^{(t+1)}(a)u^a\big|_{\theta=\theta^\star} = \mathsf{E}[b_i|Y_{\rm obs},\theta^\star] \equiv \hat{b}_i^\star,$$

$$\hat{u}_i^{(t+1)}(a)u^{2a}\big|_{\theta=\theta^\star} = \mathsf{E}[b_i^2|Y_{\rm obs},\theta^\star] \equiv \hat{\tau}_i^{2\star}$$

and $\dfrac{1}{n}\sum_{i=1}^n \hat{\tau}_i^{2\star} = \tau^{2\star}$, so we may write,

$$\widetilde{I_{\rm aug}}(a) = \qquad (\textbf{2.6.7})$$
$$\frac{1}{4\sigma^{2\star}u^{4\star}} \sum_{i=1}^n \Big[-(y_i - X_i^\top\beta)z_i\hat{b}_i^\star(a^2-2a) + 2z_i^2\hat{\tau}^{2\star}(a^2-a)\Big] - \frac{n}{2u^{4\star}}(1-a)^2.$$

Noting that all the augmented-data normal equations are satisfied for all values of a at $\theta = \theta^\star$, we evaluate the augmented-data normal equation for u^2 with $a = 1$ at $\theta^\star$ to obtain †

$$\sum_{i=1}^n z_i^2\hat{\tau}_i^{2\star} = \sum_{i=1}^n z_i\hat{b}_i^\star(y_i - X_i^\top\beta^\star). \qquad (\textbf{2.6.8})$$

Substituting (2.6.8) into (2.6.7) yields

$$\widetilde{I_{\rm aug}}(a) = \frac{1}{2u^{4\star}}\left(a^2\frac{1}{2\sigma^{2\star}}\sum_{i=1}^n z_i^2\hat{\tau}_i^2 + (1-a)^2 n\right). \qquad (\textbf{2.6.9})$$

Since $\mathsf{E}[\hat{\tau}_i^2] = u^2$, we can easily write (2.6.9) in terms of the observed quantities $\{(y_i, X_i, z_i), i = 1,\ldots,n\}$ and the maximum likelihood estimate $\theta^\star \equiv (\beta^\star, \sigma^{2\star}, u^{2\star})$ for large n. An even more satisfying result, however, is to rewrite (2.6.9) without appealing to the asymptotic behavior of $\hat{\tau}_i^2$. In order to do this we observe

† With q random effects and Δ assumed to be the identity in model (2.5.13), this expression can be generalized to $\sum_{i=1}^n Z_i^\top\hat{T}_i^\star Z_i = \sum_{i=1}^n Z_i^\top\hat{b}_i^\star(y_i - X_i^\top\beta^\star)$, where $\hat{T}_i^\star$ is (2.5.3) evaluated at $\theta^{(t)} = \theta^\star$.

$$\frac{1}{2\sigma^{2\star}}\sum_{i=1}^{n} z_i^2\hat{\tau}_i^{2\star} = \frac{1}{2\sigma^{2\star}}\sum_{i=1}^{n} z_i\hat{b}_i^{\star}(y_i - X_i^{\top}\beta^{\star})$$
$$= \frac{1}{2\sigma^{2\star}}\sum_{i=1}^{n}\frac{z_i^2u^{2\star}(y_i - X_i^{\top}\beta^{\star})^2}{\sigma^{2\star}+z_i^2u^{2\star}}$$
$$= \frac{1}{2\sigma^{2\star}}\left[\sum_{i=1}^{n}(y_i - X_i^{\top}\beta^{\star})^2 - \sum_{i=1}^{n}\frac{\sigma^{2\star}(y_i - X_i^{\top}\beta^{\star})^2}{\sigma^{2\star}+z_i^2u^{2\star}}\right] \quad (\textbf{2.6.10})$$

where the first equality is (2.6.8); the second equality follows from the first equation in (2.5.2) with $\theta^{(t)} = \theta^{\star}$ and $q = 1$ (in which case $T = u$ and $Z_i = z_i$); and the third equation follows from substituting $(z_i^2u^{2\star} + \sigma^{2\star} - \sigma^{2\star})$ for $z_i^2u^{2\star}$. To simplify (2.6.10), we use two of the observed-data normal equations $\dfrac{\partial}{\partial\theta\cdot\partial\theta^{\top}}L(\theta|Y_{\text{obs}})$, differentiating first with respect to σ^2

$$\sum_i \frac{1}{z_i^2\tau^{2\star}+\sigma^{2\star}} = \sum_i\left(\frac{y_i - X_i^{\top}\beta^{\star}}{z_i^2\tau^{2\star}+\sigma^{2\star}}\right)^2, \quad (\textbf{2.6.11})$$

and then with respect to τ^2

$$\sum_i \frac{z_i^2}{z_i^2\tau^{2\star}+\sigma^{2\star}} = \sum_i\left(\frac{z_i(y_i - X_i^{\top}\beta^{\star})}{z_i^2\tau^{2\star}+\sigma^{2\star}}\right)^2. \quad (\textbf{2.6.12})$$

Adding $\sigma^{2\star}\times$ (2.6.11) with $\tau^{2\star}\times$ (2.6.12) yields [‡]

$$n = \sum_i \frac{(y_i - X_i^{\top}\beta^{\star})^2}{z_i^2\tau^{2\star}+\sigma^{2\star}}. \quad (\textbf{2.6.13})$$

Substituting (2.6.13) into (2.6.10), we may conclude that the augmented information for u^2 given in (2.6.9) is

$$\frac{1}{2u^{4\star}}\left[a^2\left(\frac{1}{2\sigma^{2\star}}\sum_{i=1}^{n}(y_i - X_i^{\top}\beta^{\star})^2 - \frac{n}{2}\right) + (1-a)^2n\right], \quad (\textbf{2.6.14})$$

[‡] With q random effects, a similar derivation yields $n = \sum_{i=1}^{n}\dfrac{(y_i - X_i^{\top}\beta^{\star})^2}{\sigma^{2\star}+Z_i^{\top}T^{\star}Z_i}$, where T need not be diagonal.

which is minimized as a function of a for

$$a_{\text{opt}} = \left[\frac{\sum_{i=1}^{n} (y_i - X_i^\top \beta^\star)^2 / n}{2\sigma^{2\star}} + \frac{1}{2} \right]^{-1}.$$

Unfortunately, implementing the EM algorithm with augmented data $Y_{\text{aug}}(a_{\text{opt}})$ does not result in a simple closed-form M-step unless $a \in \{0, 1\}$. Thus, although such an algorithm may converge faster, it does not satisfy our objective of using algorithms that are simple and stable as well as fast. In what follows, we will therefore confine our attention to algorithms that result from $a \in \{0, 1\}$ (or, more generally, for q random effects $a \in \{0, 1\}^q$). In this class of algorithms, (2.6.14) is minimized by

$$a_{\text{opt}} = \begin{cases} 0 & \text{if } \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} (y_i - X_i^\top \beta^\star)^2 \geq 3\sigma^{2\star}, \\ 1 & \text{otherwise.} \end{cases} \qquad \textbf{(2.6.15)}$$

That is, the (small sample) augmented information $\widetilde{I_{\text{aug}}}(a)$ is smaller for the new algorithm (i.e., $a = 1$) than for the standard algorithm (i.e., $a = 0$) if and only if the total variance is less than three times the residual variance.

2.6.4. Theoretical derivations with q random effects

We now return to the general case of q random effects and will explore how $\widetilde{I_{\text{aug}}}(a)$ depends on a for $a \in \{0, 1\}^q$. We start out by generalizing $\widetilde{I_{\text{aug}}}(a)$ (e.g., (2.6.9)) to the case of q random effects. In order to derive $I_{\Delta\Delta}(a)$, we differentiate (2.6.3) twice with respect to Δ:

$$-\left. \frac{\partial^2 Q(\theta|\theta^\star)}{\partial \delta_{lm} \partial \delta_{pr}} \right|_{\theta=\theta^\star} = \frac{1}{\sigma^{2\star}} \sum_{i=1}^{n} z_{il} z_{ip} \left[\hat{U}_i^\star \right]_{rm}, \qquad \textbf{(2.6.16)}$$

where $\left[\hat{U}_i^\star\right]_{jk}$ is the jk th element of $\mathsf{E}[c_i c_i^\top | Y_{\rm obs}, \theta^\star]$, with $c_i = \Delta^{-1} b_i$. Likewise we can derive the components of $I_{\Delta U}(a)$ as

$$-\frac{\partial^2 Q(\theta|\theta^\star)}{\partial \delta_{lm} \partial u_p^2}\bigg|_{\theta=\theta^\star} = \frac{a_p}{2\sigma^{2\star} u_p^{2\star}} \sum_{i=1}^n z_{il} \left[\Delta^\star Z_i\right]_p \left[\hat{U}_i^\star\right]_{pm}, \qquad (\mathbf{2.6.17})$$

where $\left[\Delta^\star Z_i\right]_j$ is the j th component of the vector $\Delta^\star Z_i$. Finally we derive the matrix $I_{UU}(a)$ as

$$-\frac{\partial^2 Q(\theta|\theta^\star)}{\partial U \cdot \partial U}\bigg|_{\theta=\theta^\star} = \text{Diag}\left\{(1-a_1)^2 \frac{n}{2u_1^{4\star}}, \ldots, (1-a_q)^2 \frac{n}{2u_q^{4\star}}\right\} + \frac{1}{4\sigma^{2\star}} \sum_{i=1}^n D_i \hat{U}_i^\star D_i, \qquad (\mathbf{2.6.18})$$

where $D_i = \text{Diag}\left\{\frac{a_1 \left[\Delta^\star Z_i\right]_1}{u_1^{2\star}}, \ldots, \frac{a_q \left[\Delta^\star Z_i\right]_q}{u_q^{2\star}}\right\}$. The task of computing the optimal a is clearly more complicated here than when $q = 1$. In fact, we will show that, in general, we cannot order the information matrices as Theorem 2.1 requires, even for large n. The problem is somewhat simpler when T is assumed to be diagonal (i.e., we fit model (2.5.13) with Δ fixed at the identity). In this case $\widetilde{I_{\rm aug}}(a)$ reduces to (2.6.18) which can be simplified for large n by re-expressing $\hat{U}_i^\star$ using (2.5.3) evaluated at $\theta^{(t)} = \theta^\star$ (noting that $T = U$ when $\Delta = I$)

$$\hat{U}_i^\star = U^\star + \frac{U^\star Z_i Z_i^\top U^\star}{\sigma^{2\star} + Z_i^\top U^\star Z_i} \left[\frac{(y_i - X_i^\top \beta^\star)^2}{\sigma^{2\star} + Z_i^\top U^\star Z_i} - 1\right]. \qquad (\mathbf{2.6.19})$$

Thus, if $\mathsf{E}((y_i - X_i^\top \beta)^2) = \sigma^2 + Z_i^\top U Z_i$, which is certainly true if the model holds,

$$\lim_{n\to\infty} \frac{1}{n} \widetilde{I_{\rm aug}}(a) = \text{Diag}\left\{(1-a_1)^2 \frac{1}{2u_1^4} + a_1^2 \frac{\sum_{i=1}^n z_{i1}}{4n\sigma^2 u_1^2}, \ldots, (1-a_q)^2 \frac{1}{2u_q^4} + a_q^2 \frac{\sum_{i=1}^n z_{iq}}{4n\sigma^2 u_q^2}\right\}.$$

Using Theorem 2.1, when T is diagonal, the asymptotic optimal value of $a \in \{0,1\}^q$ is a such that $a_j = 0$ if and only if

$$2\sigma^2 \le \frac{u_j^2}{n} \sum_{i=1}^n \left[\Delta Z_i\right]_j^2. \qquad (\mathbf{2.6.20})$$

(We use $[\Delta Z_i]_j = z_{ij}$ here to make the (2.6.20) easier to generalize to non-diagonal T.) Condition (2.6.20) says that the standard augmentation should be used on the j th diagonal component of T if and only if the j th random effect dominates $2\sigma^2$.

Clearly, this result relies heavily on $\lim_{t\to\infty} \widetilde{I_{\text{aug}}}(a)$ being diagonal to show the augmented-data sets are nested. When T is not diagonal, the situation is more difficult since (2.6.17) involves $\frac{1}{n}\sum_i [\hat{U}_i^\star]_{jj}$ which remains positive even asymptotically. Suppose, for example, $q = 2$, $\delta_{21}^\star \neq 0$ and we choose $a' = (0,1)$ and $a = (0,0)$. It can easily be shown that $M = \lim_{n\to\infty}\left[\widetilde{I_{\text{aug}}}(a') - \widetilde{I_{\text{aug}}}(a)\right]$ then has exactly one positive and one negative eigenvalue, and thus, the augmented-data sets are not nested. Suppose further that (2.6.20) holds for $j = 2$, in which case we would expect $\widetilde{I_{\text{aug}}}(a)$ to be "smaller" than $\widetilde{I_{\text{aug}}}(a')$ and the standard EM algorithm to be faster. It turns out that whenever a and a' are of the form $(0,\ldots,0,1,\ldots,1)$, a' having exactly one less zero than a, the dominate eigenvalue of M is positive if and only if condition (2.6.20) holds (see Section 2.6.5 for details). Thus, although (2.6.20) does not assure M will be positive definite, it does indicate that it will be "more positive than negative".

We now turn our attention to the relative augmented information $R(a', a = (0,\ldots,0))$ for $a' \in \{0,1\}^q$ defined in (2.2.2). That is, we will compare each of the possible algorithms to one that uses $Y_{\text{aug}}((0,\ldots,0))$, the data augmentation that most closely approximates the standard algorithm. It is not difficult to show that

the diagonal terms of $R(a', a)$ are

$$\left\{1, \ldots, 1,\ (1-a_1)^2 + a_1^2 \frac{\sum_{i=1}^n [\Delta^\star Z_i]_1^2 \left[\hat{U}_i^\star\right]_{11}}{2n\sigma^{2\star}}, \ldots, \right.$$
$$\left. (1-a_q)^2 + a_q^2 \frac{\sum_{i=1}^n [\Delta^\star Z_i]_q^2 \left[\hat{U}_i^\star\right]_{qq}}{2n\sigma^{2\star}} \right\}$$

where the ones correspond to the elements of Δ. Since $S^{EM}(a)$ is fixed, (2.2.2) indicates that we need to minimize $R(a', a)$ in order to maximize $S^{EM}(a')$. This suggests that a good choice of data augmentation is $Y_{\text{aug}}(a')$ with

$$a'_j = \begin{cases} 0 & \text{if } 2\sigma^{2\star} \leq \dfrac{1}{n} \displaystyle\sum_{i=1}^n [\Delta^\star Z_i]_j^2 \left[\hat{U}_i^\star\right]_{jj}, \\ 1 & \text{otherwise.} \end{cases} \qquad (\textbf{2.6.21})$$

Notice that asymptotically, this condition corresponds exactly to the optimal value of a for the diagonal case given in (2.6.20).

The obvious difficulty with the conditions in (2.6.20) and (2.6.21) is that they depend on the parameter values (unlike the t-model in which the optimal algorithm is unique). Happily, however, the empirical studies showed that the standard algorithm is only slightly more efficient when $\sigma^{2\star}$ is small and that using the new algorithm always will generally lead to great computational advantage and sometimes will lead to only a slight disadvantage. Moreover, if the optimal algorithm is desired, a rough approximation of $\theta^\star$ can be used to choose an algorithm as we discussed in Section 2.6.1.

2.6.5. A dominant eigenvalue criterion for choosing an augmented-data set

Given two possible data-augmentation schemes $Y_{\text{aug}}(a)$ and $Y_{\text{aug}}(a')$ with augmented information matrices that have a positive semi-definite ordering (i.e., $M = I_{\text{aug}}(a') - I_{\text{aug}}(a)$ is either positive of negative semi-definite), Theorem 2.1 determines which will result in an EM algorithm with a faster global rate of convergence. Unfortunately, in many cases M is neither positive nor negative semi-definite. That is M will have both positive and negative eigenvalues. In such cases, we can define a dominate eigenvalue comparison of two augmented information matrices. In particular, we will say $I_{\text{aug}}(a') \geq_{\text{eigen}} I_{\text{aug}}(a)$ if the dominant eigenvalue (i.e., largest in absolute value) of M is positive. It is hoped that using this comparison in place of the positive semi-definite ordering will give a comparison that approximates the latter. As we shall see, what this comparison suggests aligns very well with the simulations presented in Section 2.6.1.

In this section we will consider the augmentations $Y_{\text{aug}}(a)$ and $Y_{\text{aug}}(a')$, where a and a' are such that $a_j = 0$ if $j \leq j_0$ and 1 otherwise; and $a'_j = 0$ for $j \leq j_0 - 1$ and 1 otherwise (for some $j_0 \leq q$). For example, with $q = 3$ and $j_0 = 2$, we would compare $a = (0, 0, 1)$ with $a' = (0, 1, 1)$. In this setting we will show that $M = \lim_{n\to\infty} \frac{1}{n}\left[\widetilde{I_{\text{aug}}}(a') - \widetilde{I_{\text{aug}}}(a)\right]$ has exactly one positive and one negative eigenvalue and that the dominant eigenvalue has the same sign as $\sum_i \frac{[\Delta^\star Z_i]^2_{j_0}}{2n\sigma^{2\star}} - \frac{1}{u^{2\star}_{j_0}}$. Thus, the asymptotic augmented information matrices do not have a positive semi-definite ordering, but they may be ordered using the dominant eigenvalue comparison. In particular, we should choose $a_j = 0$ if and only if $2n\sigma^{2\star} \leq u^\star_{j_0} \sum_i [\Delta^\star Z_i]^2_{j_0}$, which is identical to condition (2.6.20). Since the or-

der of the u_j in the information matrix is arbitrary, we may order them so that $u_j^{2\star}\sum_i [\Delta^\star Z_i]_{j_0}^2$ is decreasing in j. In this case, after the $q-1$ pairwise comparisons resulting from taking $j_0 = q, q-1, \ldots, 2$, a will be chosen so that $a_j = 0$ if and only if $2n\sigma^{2\star} \leq u_j^{2\star}\sum_i [\Delta^\star Z_i]_j^2$ for each j, again corresponding to the results in Section 2.6.4.

It remains to be shown that M has one positive and one negative eigenvalue with the aforementioned condition on the sign of the dominant eigenvalue, for which we will need the following lemma.

Lemma : Suppose the $(j+k+1)\times(j+k+1)$ real matrix M is of the form

$$M = \begin{pmatrix} 0 & \ldots & 0 & a_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & a_j & 0 & \ldots & 0 \\ a_1 & \ldots & a_j & b & c_1 & \ldots & c_k \\ 0 & \ldots & 0 & c_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & c_k & 0 & \ldots & 0 \end{pmatrix},$$

then M has one positive and one negative eigenvalue with dominant eigenvalue having the same sign as b.

Proof: It is easy to show that the characteristic function of M is

$$\lambda^{j+k-1}\left(\lambda^2 - b\lambda - \sum_{i=1}^{j} a_i^2 - \sum_{i=1}^{k} c_i^2\right)$$

and thus M has eigenvalues 0 (with multiplicity $j+k-1$) and

$$\frac{b \pm \sqrt{b^2 + 4(\sum_{i=1}^{j} a_i^2 + \sum_{i=1}^{k} b_i^2)}}{2}.$$

Since M is a real symmetric matrix, it has real eigenvalues and the result is clear.

■

From (2.6.16)–(2.6.18) it is clear that $M = \lim_{n\to\infty} \frac{1}{n}\left[\widetilde{I_{\text{aug}}}(a') - \widetilde{I_{\text{aug}}}(a)\right]$ has the form specified by the lemma, so we have proven that if a and a' are as is described above, $\widetilde{I_{\text{aug}}}(a)$ and $\widetilde{I_{\text{aug}}}(a')$ cannot be ordered in the positive semi-definite ordering sense but the dominant eigenvalue of M is positive if and only if $u_{j_0}^{2\star} \sum_{i=1}^{n} [\Delta^\star Z_i]_{j_0}^2 \geq 2n\sigma^{2\star}$.

Chapter 3

The AECM Algorithm: Model Reduction and Data Augmentation

3.1. Introduction

With the myriad of models and data structures in modern statistical analysis, maximum likelihood estimates and posterior modes are often impossible to ascertain analytically. Today's computers, however, offer enough power for many numerical optimization methods to gain popularity. In this chapter, we will examine two of these methods, data augmentation in the EM algorithm and model reduction in the CM algorithm and will show how they can be combined to produce simple, stable and efficient algorithms.

As we recall, the EM algorithm augments the observed data, $Y_{\rm obs}$, to the larger augmented-data set $Y_{\rm aug}$. It then computes the maximum likelihood estimate $\theta^\star$ as the convergent value of the iteration which sets $\theta^{(t+1)}$ to the maximizer of $Q(\theta|\theta^{(t)}) = \mathsf{E}\left[L(\theta|Y_{\rm aug})|Y_{\rm obs},\theta^{(t)}\right]$. The idea is to select $Y_{\rm aug}$ so that $\theta^{(t+1)}$ is easy to compute, thereby providing a simple, stable, although sometimes slow algo-

rithm. The CM algorithm, on the other hand, starts with a set of $S \geq 1$ (vector) constraint functions $G = \{g_s(\theta), s = 1, \ldots, S\}$ that are "space filling" (Meng and Rubin, 1993) in the sense of allowing maximization over the entire parameter space. The algorithm then sets $\theta^{(t+\frac{s}{S})}$ to the maximizer of $L(\theta|Y_{\text{obs}})$, subject to the constraint $\theta \in \Theta_s \equiv \{\theta \in \Theta : g_s(\theta) = g_s(\theta^{(t+\frac{s-1}{S})})\}$ for $s = 1, \ldots, S$. This is repeated until the algorithm converges to $\theta^\star$. The idea is to choose G so that each of the constrained maximizations is easy to implement so as to again produce a simple algorithm. One common choice of G is the partition $\{g_s(\theta) = \vartheta_{2s}, s = 1, \ldots, S\}$, where ϑ_{2s} is a subvector of $\theta = (\vartheta_{1s}, \vartheta_{2s}), s = 1, \ldots, S$.

Both the ECM (Section 1.3) and SAGE (Section 1.5) algorithms combine data augmentation and model reduction. ECM starts with EM and adds model reduction and SAGE starts with CM and adds data augmentation. In particular, the ECM algorithm starts with EM in cases where although the M-step is not in closed form, there is a space-filling set of constraints G, such that $Q(\theta|\theta^{(t)})$ is simple to maximize subject to each of the constraints. In such cases, ECM replaces the M-step of EM with the CM-steps defined by G, that is, one iteration of the CM algorithm.

The SAGE algorithm, on the other hand, starts with a CM algorithm in which each of the constraints in G form a partition of θ, but even the constrained maximizations are not all in closed form. In this case, SAGE uses data augmentation (i.e., EM) to optimize $L(\theta|Y_{\text{obs}})$ over Θ_s. That is, if a CM-step is not in closed form, SAGE replaces it with one EM iteration of an algorithm that is designed to iteratively calculate the maximizer of $L(\theta|Y_{\text{obs}})$ over Θ_s. We interpret SAGE in

this way to highlight the fact that since each EM algorithm maximizes $L(\theta|Y_{\text{obs}})$ over a different subspace of Θ, it is natural to adopt a different data-augmentation scheme for each.

The SAGE algorithm allows for more general data augmentation then does ECM in that a different data augmentation can be used at each CM-step. On the other hand, the ECM algorithm is more general than SAGE in that the constraint functions need not partition θ. Moreover, the MCECM algorithm (a slight generalization of ECM discussed in Section 1.3) allows additional E-steps to be added to each ECM iteration, which consists of an E-step followed by S CM-steps, while the SAGE algorithm must have an E-step preceding each of S CM-step. Thus, ECM and SAGE generalize EM and CM in different directions. In what follows, we will develop an algorithm that combines data augmentation and model reduction in a way that is more general than either ECM or SAGE. The Alternating Expectation/Conditional Maximization or AECM algorithm is built on the intuition of Dempster, Laird, and Rubin (1977) but, as with ECM, incorporates model reduction to simplify the M-step; takes advantage of more flexible data augmentation as in SAGE; and allows some permutation of the component steps within each iteration. In Section 3.2 we will present the details of the AECM algorithm followed by several convergence results in Section 3.3. Section 3.4 presents an example which demonstrates how the combination of data augmentation and model reduction in AECM can be used to develop algorithms that are not only simple and stable, but also fast.

3.2. The AECM Algorithm

To present our algorithm in its most general form, we need to extend and standardize the indexing system that has been commonly used in the EM literature. Specifically, we need to develop the concept of a "cycle" in-between a "step" and an "iteration."

Definition 3.1: A *cycle* consists of an E-step followed by a ordered set of CM-steps, the last of which will be followed immediately by a new E-step (which is itself the beginning of the next cycle). An *iteration* consists of one or more cycles.

For example, in the ECM algorithm, an iteration is the same as a cycle, but for MCECM, an iteration consists of multiple cycles (hence its name). In what follows, we will use t, c, and s to index i**t**eration, **c**ycle, and **s**tep, respectively, as illustrated in the following diagram.

$$\underbrace{\underbrace{\text{E}-\text{step},\ \{\text{CM}_s\ \text{step}, s=1,\ldots,S_{c_1}\}}_{[c_t+1]\text{th cycle}},\ldots,\underbrace{\text{E}-\text{step},\ \{\text{CM}_s\ \text{step}, s=1,\ldots,S_{c_2}\}}_{[c_{t+1}]\text{th cycle}}}_{t\text{th iteration}}$$

The flexibility of our general algorithm comes from allowing the data-augmentation scheme, as well as the set of constraint functions, to depend on the cycle index. Because of the alternating nature of the E-step as a consequence of the changing data-augmentation scheme, we call our algorithm the Alternating Expectation/Conditional Maximization or AECM algorithm. In the general framework, for

cycle c we write $Y_{\text{aug}}^{[c]} = (Y_{\text{obs}}, Y_{\text{mis}}^{[c]})$, where $Y_{\text{mis}}^{[c]}$ is the unobserved part of $Y_{\text{aug}}^{[c]}$. For a given c, we select a set of S_c (vector) functions of θ, $G_c = \{g_s^{[c]}(\theta), s = 1, \ldots, S_c\}$ which induces a sequence of subspaces $\{\Theta_s^{[c]}, s = 1, \ldots, S_c\}$ of the parameter space Θ, where $\Theta_s^{[c]} = \{\theta \in \Theta : g_s^{[c]}(\theta) = g_s^{[c]}(\theta^{\left[c+\frac{s-1}{S_c}\right]})\}$. The $[c+1]st$ cycle then consists of

E-step: Compute $Q_{c+1}(\theta|\theta^{[c]}) = \int L(\theta|Y_{\text{aug}}^{[c+1]}) f(Y_{\text{mis}}^{[c+1]}|Y_{\text{obs}}, \theta^{[c+1]}) dY_{\text{mis}}^{[c+1]}$

and S_{c+1} CM-steps

s th CM-step: Calculate $\theta^{\left[c+\frac{s}{S_{c+1}}\right]}$ such that

$$Q_{c+1}(\theta^{\left[c+\frac{s}{S_{c+1}}\right]}|\theta^{[c]}) \geq Q_{c+1}(\theta|\theta^{[c]}) \quad \text{for} \quad \text{all} \quad \theta \in \Theta_s^{[c+1]}, s = 1, \ldots S_{c+1}. \quad \textbf{(3.2.1)}$$

The input of the next cycle is taken as $\theta^{\left[c+\frac{S_{c+1}}{S_{c+1}}\right]} = \theta^{[c+1]}$.

It is clear that without proper restrictions on the constraint functions, $G^{[c]} = \{g_s^{[c]}, s = 1, \ldots, S_c\}$, there is no reason to hope that the AECM algorithm will converge properly. The needed condition here is the *space-filling* condition which Meng and Rubin (1993) used for the ECM algorithm. Intuitively, this condition requires that, after a set of constrained maximizations, we will have searched in all directions (radiating from a particular origin) of the parameter space. Operationally, the space-filling condition holds for $G^{[c]} = \{g_s^{[c]}, s = 1, \ldots, S_c\}$ at a particular $\theta' \in \Theta$ if and only if (see Meng and Rubin, 1993)

$$\bigcap_{s=1}^{S_c} J_s^{[c]}(\theta') = \{0\}, \qquad \textbf{(3.2.2)}$$

where

$$J_s^{[c]}(\theta) = \left\{\nabla g_s^{[c]}(\theta')\lambda : \lambda \in \mathbb{R}^{d_s^{[c]}}\right\}$$

is the column space of the gradient of the $d_s^{[c]}$-dimensional vector $g_s^{[c]}(\theta)$. (We always assume $g_s^{[c]}(\theta)$ is differentiable and $\nabla g_s^{[c]}(\theta)$ is of full rank at each interior point of Θ to avoid unnecessary technical complications.) For the ECM algorithm, each iteration consists of one cycle, and $G^{[c]}$ does not depend on the cycle index. Thus, it was sufficient in Meng and Rubin (1993) to only consider (3.2.2) for one cycle. With AECM, however, it is possible that the space-filling condition will not hold for every cycle, that is, it may be useful (for the efficiency of the algorithm) to allow the space-filling condition to be satisfied only after several cycles. More precisely, in contrast with (3.2.2), we now only require

$$\bigcap_{c=c_1}^{c_2} \bigcap_{s=1}^{S_c} J_s^{[c]}(\theta') = \{0\}. \qquad (\mathbf{3.2.3})$$

For theoretical as well as practical reasons, we will define an iteration of the AECM algorithm as the smallest set of consecutive cycles such that (3.2.3) holds. More precisely, we define an AECM iteration sequence $\{\theta^{(t)}, t \geq 0\}$ † as a subsequence of the sequence generated by the output of each cycle $\{\theta^{[c]}, c \geq 0\}$ such that $\theta^{(t+1)} = \theta^{[c_{t+1}]}$ if

$$\bigcap_{c=c_t+1}^{c_{t+1}} \bigcap_{s=1}^{S_c} J_s^{[c]}(\theta^{[c_t]}) = \{0\}, \text{ but } \bigcap_{c=c_t+1}^{c_{t+1}-1} \bigcap_{s=1}^{S_c} J_s^{[c]}(\theta^{[c_t]}) \neq \{0\}. \qquad (\mathbf{3.2.4})$$

In other words, we consider a set of consecutive cycles to form an iteration of the AECM algorithm if and only if the last cycle of the set has completed the search of the parameter space, starting from the previous iteration (not cycle) output, in the sense of completing the space-filling requirement. Since the cycles are time-ordered,

† For clarity, we use parenthesis in the superscript indexing iteration number and square brackets in the superscript indexing cycle number.

there is a unique sequence $\{c_t; t \geq 0\}$ that defines the iteration sequence. To avoid the pathological theoretical possibility that there may be only a finite number of c_t which satisfy (3.2.4), we assume the sequence $\{c_t; t \geq 0\}$ contains infinitely many distinct numbers.

There are three reasons that call for this definition of an iteration rather than defining iterations as cycles (e.g., as in Hero and Fessler, 1994). First, the common notion of an iteration *completely* updates θ, and thus it is natural to require that the whole space be searched at each iteration. (Note that even if a cycle changes the values of all the components of θ, it may not completely update θ, as a reparameterization of θ will reveal if the space-filling condition is not satisfied.) Second, it is theoretically easier and more satisfactory to study an iteration mapping from $\theta^{(t)}$ to $\theta^{(t+1)}$ when the mapping does not depend on t, which for algorithms that satisfy the space-filling condition, is only possible (in addition to other requirements) when each of the iterations searches the whole parameter space. Finally, when monitoring convergence, which is of paramount concern in practice, the iterates must be comparable. In particular, it is only meaningful to monitor the difference of consecutive iterates (or functions thereof) when the iterates each represent a complete update of the parameter rather than just part of it (in which case one can easily be misled and declare convergence when a small difference is caused by a small partial update of the parameter).

Table 3.1 provides an overview of how AECM generalizes several existing algorithms including EM, ECM, SAGE, and ECME (which is given special attention in Section 3.3.2). In addition to these, there are useful implementations of the

AECM	$Y_{\text{aug}}^{[c]}$	C_t	S_c	G_c
EM	Y_{aug}	1	1	none
ECM	Y_{aug}	1	S	G
PECM	Y_{aug}	1	S	$\{g_s^{[c]}(\theta) = \vartheta_s\}$
MCECM	Y_{aug}	C	1	$G_c = G_{c \bmod C}$
ECME	$\begin{cases} Y_{\text{aug}} & c \text{ odd} \\ Y_{\text{obs}} & c \text{ even} \end{cases}$	2	$\begin{cases} S_1 & c \text{ odd} \\ S_2 & c \text{ even} \end{cases}$	$\begin{cases} G_1 & c \text{ odd} \\ G_2 & c \text{ even} \end{cases}$
SAGE	$Y_{\text{aug}}^{[c]}$	C_t	1	$\{g_s^{[c]}(\theta) = \vartheta_s\}$

Table 3.1. The special cases of the AECM algorithm. The table records the data-augmentation scheme $(Y_{\text{aug}}^{[c]})$, number of cycles per iteration (C_t), number of CM-steps per cycle (S_c), and constraint functions (G_c). When the index c (or t) is suppressed in the table, the quantity is fixed between cycles (or iterations). The ECM algorithm introduced model reduction to EM via the S constraint functions in G. In PECM these constraints have a special form which partitions θ, that is ϑ_s is a sub-vector of θ. The multi-cycle ECM (MCECM) algorithm adds an E-step before each of the CM-steps. (All of these E-steps need not be added, but then the notation is more cumbersome.) The ECME algorithm is actually more general than is presented here in that the $S_1 + S_2$ CM-steps need not be separated into two cycles but can be performed in any order. The SAGE algorithm introduced variable data-augmentation, but updated only one sub-vector of θ at a time. In Fessler and Hero (1994) an iteration is equivalent to our cycle.

AECM algorithm which are not instances of any of the special cases in Table 3.1. These include a generalized version of the optimal algorithm of Section 2.3 for fitting multivariate t-distributions with unknown degrees of freedom, which will be discussed in Section 3.4.

3.3. Convergence Theorems

3.3.1. Convergence of AECM

We now proceed to show that a sequence $\{\theta^{(t)}\}$ of iterates from an AECM algorithm increases $L(\theta|Y_{\text{obs}})$ at each iteration and that under standard regularity conditions, AECM algorithms converge to a stationary point of $L(\theta|Y_{\text{obs}})$. These results are the counterparts of the results for EM (Dempster, Laird, and Rubin, 1977; Wu, 1983) and for ECM (Meng and Rubin, 1993), and provide more complete results for ECME and SAGE.

Our first result, which is most fundamental, states that AECM, like all of its predecessors, maintains monotonic convergence of the likelihood values. Note that this result dose not require the space-filling condition.

Theorem 3.1: Any AECM sequence increases (or maintains) $L(\theta|Y_{\text{obs}})$ at every cycle and thus increases (or maintains) $L(\theta|Y_{\text{obs}})$ at every iteration.

Proof: The result follows trivially from

$$Q_{c+1}(\theta^{\left[c+\frac{s}{S_{c+1}}\right]}|\theta^{[c]}) \geq Q_{c+1}(\theta^{\left[c+\frac{s-1}{S_{c+1}}\right]}|\theta^{[c]}), \quad \text{for} \quad s = 1, \ldots, S_{c+1}$$

(as a consequence of (3.2.1)) and the Jensen inequality as used in Dempster, Laird and Rubin (1977).

■

In order to prove the algorithm converges to $\theta^\star \in \mathcal{L} = \{\theta \in \Theta_0 : \frac{\partial L(\theta|Y_{\text{obs}})}{\partial \theta} = 0\}$ where Θ_0 is the interior of Θ (i.e., $\theta^\star$ is an interior stationary point), several regularity conditions similar to the standard ones used by Wu (1983) and Meng

and Rubin (1993) for EM and ECM respectively will be required, in particular, we assume Wu's (1983) conditions (6) – (10). This is formalized by the global convergence theorem (c.f., Wu, 1983, Zangwill, 1969) which states that it suffices to show

(i) The points $\{\theta^{(t)}\}$ are contained in a compact set of Θ;

(ii) The AECM mapping, $\theta^{(t+1)} = M^{AECM}(\theta^{(t)})$, is closed;

(iii) $L(\theta^{(t+1)}|Y_{\text{obs}}) \geq L(\theta^{(t)}|Y_{\text{obs}})$ with equality only if $\theta^{(t)} \in \mathcal{L}$.

The weak inequality in (iii) is an immediate consequence of Theorem 3.1. This, along with the assumption that the set $\Theta_{\theta^{(0)}} = \{\theta \in \Theta : L(\theta|Y_{\text{obs}}) \geq L(\theta^{(0)}|Y_{\text{obs}})\}$ is compact for any $L(\theta^{(0)}|Y_{\text{obs}}) > -\infty$ (i.e., Wu's condition (6)), gives (i). Thus, it remains only to show (ii) and the equality statement of (iii) hold.

Under Wu's compactness condition (6) and continuity condition (10), the same argument used in Meng and Rubin (1993) to show that the ECM mapping, M^{ECM}, is closed can be applied to establish that the mapping determined by each AECM cycle (i.e., $\theta^{[c+1]} = M^{AECM}_{[c]}(\theta^{[c]})$) is closed. Since the iteration mapping $\theta^{(t+1)} = M^{AECM}_{(t)}(\theta^{(t)})$ is a composition of several cycle mappings, it is also closed. Unlike the ECM algorithm where M^{ECM} does not change with iteration, however, an AECM mapping can vary with iteration, and thus, Zangwill's (1969) global convergence theorem does not apply directly. Although it is possible to extend Zangwill's global convergence theorem to iterate-dependent mappings by imposing additional regularity conditions, we take a simpler approach by directly requiring $M^{AECM}_{(t)}(\theta)$ not depend on t for $t \geq t_0$ (typically $t_0 = 1$), and hence the simplified notation given in (ii). The reason we choose to restrict $M^{AECM}_{(t)}(\theta)$ in this way

is that, although the data-augmentation scheme in AECM varies with cycle and thus with iteration, the resulting iteration mapping typically remains the same at each iteration in real applications, as shall be illustrated by our t-model example in Section 3.4. If useful applications of AECM arise in which $M_{(t)}^{AECM}$ depends on t, we will develop the more general theory accordingly.

Finally, the equality statement of (iii) can be established using the space-filling condition in the same way as in the proof of Theorem 2 of Meng and Rubin (1993) because each AECM iteration contains a complete search of the parameter space. The only modification needed is to extend (4.5) of Meng and Rubin (1993) to cover all the cycles contained in one AECM iteration (since ECM contains only one cycle per iteration) and to note that their (4.3) holds for any $Q_c(\theta|\theta^{[c]}),\ c \geq 1$. This completes the proof of the following main result on the convergence of an AECM iteration sequence $\{\theta^{(t)}, t \geq 0\}$.

Theorem 3.2: Suppose (a) all the conditional maximizations in (3.2.1) are unique and (b) the AECM iteration mapping, $M_{(t)}^{AECM} : \theta^{(t)} \rightarrow \theta^{(t+1)}$ does not depend on t, then all the limit points of a AECM iteration sequence $\{\theta^{(t)}, t \geq 0\}$ are stationary points of $L(\theta|Y_{\text{obs}})$.

As discussed in Meng and Rubin (1993), the uniqueness condition (a) is often satisfied in practice (e.g., when the conditional maximizations are in closed form), but even this condition can be eliminated if we force $\theta^{[c+\frac{s}{S_{c+1}}]} = \theta^{[c+\frac{s-1}{S_{c+1}}]}$ whenever there is no increase in $Q_{c+1}(\theta|\theta^{[c]})$ at the s th CM-step within the $(c+1)$ st cycle. Other conditions are also possible to ensure the result as discussed in Meng and Rubin (1993). Corollary 1 of Meng and Rubin (1993) also holds here,

that is, if $L(\theta|Y_{\text{obs}})$ is unimodal with $\theta^\star$ being the only stationary point then under the conditions of Theorem 3.2, any AECM iteration sequence will converge to the global maximizer starting from any $\theta^{(0)} \in \Theta_0$.

Finally, if in addition to assuming that $M_{(t)}^{AECM}$ does not depend on t, we assume each iteration has the same set of cycles, say $\{c_1, \ldots, c_C\}$, we have the following result regarding the rate of convergence of AECM, the proof of which is essentially the same as Meng's (1994) proof of the rate of convergence for the multi-cycle ECM algorithm.

Theorem 3.3: Suppose the AECM iteration mapping is a composition of C fixed-cycle mappings, all the conditional maximizations in (3.2.1) satisfy the Lagrange Multiplier equations, and $\theta^{\left[c+\frac{s}{S_{c+1}}\right]} \to \theta^\star$ as $c \to \infty$. Then the (matrix) rate of convergence of the AECM iteration is

$$DM^{AECM} = \prod_{c=1}^{C} \left\{ I_{\text{mis}}^{[c]} \left[I_{\text{aug}}^{[c]} \right]^{-1} + \left(I_{\text{obs}} \left[I_{\text{aug}}^{[c]} \right]^{-1} \right) \prod_{s=1}^{S_c} P_s^{[c]} \right\} \qquad (\mathbf{3.3.1})$$

where

$$I_{\text{mis}}^{[c]} = \int -\frac{\partial^2 \log f(Y_{\text{mis}}^{[c]}|Y_{\text{obs}}, \theta)}{\partial\theta \cdot \partial\theta^\top} f(Y_{\text{mis}}^{[c]}|Y_{\text{obs}}, \theta) dY_{\text{mis}}^{[c]} \Big|_{\theta=\theta^\star},$$

$$I_{\text{aug}}^{[c]} = \int -\frac{\partial^2 \log f(Y_{\text{aug}}^{[c]}|\theta)}{\partial\theta \cdot \partial\theta^\top} f(Y_{\text{mis}}^{[c]}|Y_{\text{obs}}, \theta) dY_{\text{mis}}^{[c]} \Big|_{\theta=\theta^\star},$$

I_{obs} is given in (1.6.5), and $P_s^{[c]} = \nabla_s^{[c]} \left[[\nabla_s^{[c]}]^\top \left[I_{\text{aug}}^{[c]} \right]^{-1} \nabla_s^{[c]} \right]^{-1} [\nabla_s^{[c]}]^\top \left[I_{\text{aug}}^{[c]} \right]^{-1}$ with $\nabla_s^{[c]} = \nabla g_s^{[c]}(\theta^\star)$, $\prod_{s=1}^{S_c} P_s^{[c]} = P_1^{[c]} P_2^{[c]} \cdots P_{S_c}^{[c]}$, $G_c = \{g_s^{[c]}(\theta), s = 1, \ldots, S_c\}$, and $Y_{\text{aug}}^{[c]}$ being determined by the cth cycle, $c = 1, \ldots, C$. The global rate of convergence is governed by the spectral radius of DM^{AECM}.

When $C = 1$, the supplemented ECM algorithm (developed in Chapter 4) uses (3.3.1) to calculate the asymptotic variance-covariance matrix of $\theta^\star$, I_{obs}^{-1}

as a function of DM^{ECM}, $\prod_{s=1}^{S_k} P_s$ and I_{aug}. When $C > 1$, there is often a corresponding algorithm with $C = 1$ which may be less efficient (e.g., SAGE) for estimating $\theta^\star$ but can be used in conjunction with SECM to calculate I_{obs}^{-1}, once $\theta^\star$ has been obtained. This will be further discussed in Section 4.5.2. The relationship between the spectral radius of DM, the largest eigenvalue of DM, and the global rate of convergence will be taken up in Section 5.2.

Theorem 3.3 assumes $\theta^{\left[c+\frac{s}{S_{t+1}}\right]} \to \theta^\star$, whereas Theorem 3.2 only assures that $\theta^{(t)} \to \theta^\star$. In order to help bridge this theoretical gap we can show that $\{\theta^{[c]}, c \geq 0\}$ converges to $\theta^\star$ along with its subsequence $\{\theta^{(t)}, t \geq 0\}$ if, in addition to the assumptions of Theorem 3.2, we assume that for $c \geq c'$, $\theta^{[c]} \in \Theta'$, a compact subset of Θ such that $L(\theta^\star|Y_{\text{obs}}) \geq L(\theta'|Y_{\text{obs}})$ for all $\theta' \in \Theta'$. This result follows directly from the following lemma.

Lemma S: Suppose that U is a compact set and $f : U \to \mathbb{R}$ is a continuous function with a unique global maximum at $x^\star$, so that $f(x) < f(x^\star)$ for any $x \in U, x \neq x^\star$. Furthermore, let $\{x_i\}_{i=1}^\infty$ be a sequence of points in U such that the sequence $\{f(x_i)\}_{i=1}^\infty$ is nondecreasing and such that there exists a convergent subsequence $\{x_{n_i}\}_{i=1}^\infty$ having $x^\star$ as its limit. Then the entire sequence $\{x_i\}_{i=1}^\infty$ converges to $x^\star$.

Proof: We first prove that given any $\epsilon > 0$ there exists $\delta > 0$ such that $|f(x) - f(x^\star)| < \delta$ implies $|x - x^\star| < \epsilon$. Suppose otherwise, then, for any natural number N we could find an $x_N \in U$ satisfying $|f(x_N) - f(x^\star)| < \frac{1}{N}$ while $|x_N - x^\star| \geq \epsilon$. In this way we obtain a sequence $\{x_N\}_{N=1}^\infty$ with each $x_N \in U - B_\epsilon(x^\star)$, which is also a compact set (where $B_\epsilon(x^\star)$ is the open ball of radius ϵ and center $x^\star$).

Therefore, there exists a convergent subsequence $\{x_{N_j}\}_{j=1}^{\infty}$ whose limit we shall denote $\tilde{x}$. Therefore

$$f(\tilde{x}) = f(\lim_{j\to\infty} x_{N_j}) = \lim_{j\to\infty} f(x_{N_j}) = f(x^\star),$$

where we have used the continuity of f in the middle step. However, $\tilde{x} \neq x^\star$ since $|\tilde{x} - x^\star| \geq \epsilon$ by construction, which contradicts the hypothesis that $x^\star$ was the unique point at which f was maximized on U.

So suppose that $\epsilon > 0$ is given. By the above argument we can find $\delta > 0$ such that $|f(x) - f(x^\star)| < \delta$ implies $|x - x^\star| < \epsilon$. Since the subsequence $\{x_{n_i}\}_{i=1}^{\infty}$ converges to $x^\star$, the sequence $\{f(x_{n_i})\}_{i=1}^{\infty}$ converges to $f(x^\star)$ since f is continuous. In particular, we can choose M large enough so that $i \geq M$ implies $|f(x_{n_i}) - f(x^\star)| < \delta$. As $f(x^\star)$ is a maximum this means that $f(x^\star) - \delta < f(x_{n_i}) \leq f(x^\star)$. But by assumption, the entire sequence $\{f(x_i)\}_{i=1}^{\infty}$ is increasing, so once one of the terms is within δ of $f(x^\star)$, they all must be. In other words, $|f(x_i) - f(x^\star)| < \delta$ for all $i \geq n_M$, which means that $|x_n - x^\star| < \epsilon$, using our result from the previous paragraph. Since ϵ can be chosen arbitrarily small, we have proven that

$$\lim_{i\to\infty} x_i = x^\star.$$

■

3.3.2. A note on step ordering within ECME

The ECME algorithm takes advantage of a simple idea in order to increase the rate of convergence of the ECM algorithm. The CM-steps of ECM maximize $Q(\theta|\theta^{(t)})$

under each of the constraints in G in turn. When it is computationally attractive, Liu and Rubin (1995a) replace several of these CM-steps with conditional maximizations of $L(\theta|Y_{\text{obs}})$ (using the same constraint functions) and accomplish a remarkable increase in the rate of convergence. Unfortunately, there is a technical error in their proof that an ECME algorithm monotonically increases the likelihood, which relies on the assertion that the likelihood is increased *at each* CM-step. This clearly does not always hold. For an extreme example, consider an ECME algorithm that, after calculating $Q(\theta|\theta^{(t)})$ in the E-step performs one unconstrained maximization of $L(\theta|Y_{\text{obs}})$ followed by one unconstrained maximization of $Q(\theta|\theta^{(t)})$. Clearly, the output from the second CM-step can decrease $L(\theta|Y_{\text{obs}})$. The problem stems from the statement in their proof that if $Q(\theta|\theta^{(t)})$ increases, so does $L(\theta|Y_{\text{obs}})$ (by Jensen's inequality), that is

$$Q(\theta|\theta^{(t)}) \geq Q(\tilde{\theta}|\theta^{(t)}) \Longrightarrow L(\theta|Y_{\text{obs}}) \geq L(\tilde{\theta}|Y_{\text{obs}}), \qquad (\mathbf{3.3.2})$$

which is, in fact, only guaranteed when $\tilde{\theta} = \theta^{(t)}$.

The AECM algorithm avoids this problem by always calculating $\theta^{[c+1]}$ such that $Q_{c+1}(\theta^{[c+1]}|\theta^{[c]}) \geq Q_{c+1}(\theta^{[c]}|\theta^{[c]})$ via a series of CM-steps, each of which increase $Q_{c+1}(\theta|\theta^{[c]})$. If it is computationally advantageous, $L(\theta|Y_{\text{obs}})$ can then be increased directly via a set of CM-steps in a separate cycle. Put another way, we always perform an E-step whenever we introduce a different data augmentation.

It should be noted that, although it is easy to find examples of ECME algorithms that do not increase $L(\theta|Y_{\text{obs}})$ at each CM-step, we have not yet found an example of an ECME algorithm that does not increase $L(\theta|Y_{\text{obs}})$ at each iteration.

It is a proof that ECME (with arbitrary CM-step order) increases the likelihood at each iteration that we are without.

3.4. Example: Fitting the t-Model with Unknown df

In Section 2.3 we presented an algorithm for fitting the multivariate t-distribution which is more efficient than the standard iteratively reweighted least squares algorithm when the degrees of freedom are known. Here, we will consider the same model but with unknown degrees of freedom, a problem that is also of practical interest (e.g., Lange, Little, and Taylor, 1989). The standard EM algorithm (i.e., using the augmentation $Y_{\text{aug}} = \{(y_i, q_i), i = 1, \ldots, n\}$ as described in (2.3.1)) can be very slow to converge, as Liu and Rubin (1995b) illustrated in their presentation of the ECME algorithm, which offers great computational gain in this problem. In this section, we will both extend the optimal algorithm of Section 2.3 to estimate ν and combine it with the ECME algorithm. We will investigate the performance of the resulting algorithms empirically and show that the extension of the optimal algorithm can outperform both the standard algorithm and ECME.

In order to construct the algorithm, we first reduce the model by breaking the parameter space into two parts in order to use a two cycle algorithm. In the first cycle, we will update (μ, Σ) given ν, and in the second cycle, we will update ν given (μ, Σ). In other words, we choose $\Theta^{[c]} = \{\theta \equiv (\mu, \Sigma, \nu) \in \Theta : \nu = \nu^{[c-1]}\}$

for c odd and $\Theta^{[c]} = \{\theta \in \Theta : (\mu, \Sigma) = (\mu^{[c-1]}, \Sigma^{[c-1]})\}$ for c even. The resulting AECM algorithm has two cycles in each iteration and one CM-step in each cycle as illustrated below. (We postpone an explicit formulation of $Q_{c+1}(\theta|\theta^{[c]})$ and $Q_{c+2}(\theta|\theta^{[c+1]})$ for the moment.) The generic cycle index c here is assumed to be even.

Odd numbered cycles:

E-step: Compute $Q_{c+1}(\theta|\theta^{[c]})$

CM-step: Calculate $\mu^{[c+1]}$ and $\Sigma^{[c+1]}$ such that

$$Q_{c+1}(\theta^{[c+1]} \equiv (\mu^{[c+1]}, \Sigma^{[c+1]}, \nu^{[c]})|\theta^{[c]}) \geq Q_{c+1}(\theta|\theta^{[c]})$$

for all θ such that $\nu = \nu^{[c]}$.

Even numbered cycles:

E-step: Compute $Q_{c+2}(\theta|\theta^{[c+1]})$

CM-step: Calculate $\nu^{[c+2]}$ such that

$$Q_{c+2}(\theta^{[c+2]} \equiv (\mu^{[c+1]}, \Sigma^{[c+1]}, \nu^{[c+2]})|\theta^{[c+1]}) \geq Q_{c+2}(\theta|\theta^{[c+1]})$$

for all θ such that $\mu = \mu^{[c+1]}$ and $\Sigma = \Sigma^{[c+1]}$.

The iterations are defined by $\theta^{(t)} = \theta^{[2t]}$, and the space-filling condition is easily satisfied by this algorithm because $G_1 \cup G_2$ partition Θ as in the PECM algorithm and each iteration uses the same partition $G_1 \cup G_2$.

As was discussed in Section 2.3, there are two choices for the data augmentation when ν is known (i.e., with the odd numbered cycles) which result in CM-steps that are simple to compute. The first is the standard aug-

mentation $\{(y_i, q_i), i = 1, \ldots, n\}$, and the second is the optimal augmentation $\{(y_i, q_i(a_{\text{opt}}^{[c]})), i = 1, \ldots, n\}$, where (y_i, q_i) is as in model (2.3.1), $q_i(a) = |\Sigma|^{-a} q_i$, and $a_{\text{opt}}^{[c]} = 1/(\nu^{[c]} + p)$. We see here that since $\nu^{[c]}$ changes with c, the optimal data augmentation is a function of the cycle, and thus, the resulting algorithm does not fit into the standard EM (or ECM) paradigm. There are also two data-augmentation schemes when we condition on μ and Σ (i.e., with the even numbered cycles): the standard augmentation, $\{(y_i, q_i), i = 1, \ldots, n\}$, and no augmentation, $\{y_i, i = 1, \ldots, n\}$. The latter was used by Liu and Rubin (1995a) in their ECME implementation. Since the working parameter a does not affect the cycle for updating $\nu^{[c]}$, the optimal and standard augmentation are equivalent when c is even. There are no known data-augmentation schemes that result in a closed-form update of ν, and both of these data augmentations require similar computations via a univariate optimization routine.

In conjunction with the standard data augmentation in the odd cycles the two augmentation schemes in the even cycles result in the MCECM and ECME algorithms, respectively, and were compared by Liu and Rubin (1995a and 1995b). Here we compare MCECM and ECME with two new algorithms (AECM 1 and AECM 2, respectively) which result from replacing the standard augmentation with the optimal augmentation when updating (μ, Σ) (see Table 3.2). As we shall see, our simulations indicate that in order to achieve fast algorithms, the choice of data augmentation when updating (μ, Σ) is more critical than when updating ν.

Implementation of all four algorithms is straightforward. The odd-numbered cycles are conditional on the current iterate $\nu^{[c]}$ and are implemented exactly as

Model Reduction (CM-steps):	update μ and Σ		update ν	
Data-Augmentation (E-step):	$\{(y_i, q_i)\}$	$\{(y_i, q_i(a_{\text{opt}}^{(t)})\}$	$\{(y_i, q_i)\}$	$\{(y_i)\}$
MCECM	$\times$		$\times$	
ECME	$\times$			$\times$
AECM 1		$\times$	$\times$	
AECM 2		$\times$		$\times$

Table 3.2. AECM algorithms used to fit the multivariate t with unknown degrees of freedom. When the degrees of freedom are unknown, it is convenient to first update (μ, Σ) conditional on ν and then update ν conditional on (μ, Σ). The MCECM algorithm computes the conditional expectation of the augmented-data set $\{(y_i, q_i), i = 1, \ldots, n\}$ in an E-step before each of the CM-steps. The ECME algorithm uses only the observed data to update ν, and the extension of the optimal EM algorithm of Section 2.3 (AECM 1) uses the optimal data-augmentation $\{(y_i, q_i(a_{\text{opt}}^{(t)}), i = 1, \ldots, n\}$ when updating (μ, Σ). Using both of these replacements simultaneously results in the algorithm referred to as AECM 2.

described in Section 2.3 with ν replaced by $\nu^{[c]}$. With the change of notation from $(t+1)$ to $[c+1]$, the E-step is given in (2.3.2), and for the standard augmentation, the CM-step is given in (2.3.3) and (2.3.4). For the optimal augmentation we replace (2.3.4) with (2.3.6). Regardless of the data augmentation, the even cycles require numerical optimization of $Q_{c+2}(\theta|\theta^{[c+1]})$, where c is even. Specifically, when using $Y_{\text{aug}} = \{(y_i, q_i), i = 1, \ldots, n\}$, we first execute an E-step which sets $w_i^{[c+2]} = (\nu^{[c]} + p)/(\nu^{[c]} + d_i^{[c+1]})$ for each i, and then set $\nu^{[c+2]}$ equal to the solution of the equation:

$$-\phi\left(\frac{\nu}{2}\right)+\log\left(\frac{\nu}{2}\right)+\phi\left(\frac{\nu^{[c]}+p}{2}\right)-\log\left(\frac{\nu^{[c]}+p}{2}\right)$$
$$+\frac{1}{n}\sum_i\left[\log w_i^{[c+2]}-w_i^{[c+2]}\right]+1=0, \qquad (\mathbf{3.4.1})$$

where $\phi(\cdot)$ is the digamma function. Likewise, when using $Y_{\text{aug}} = \{y_i, i = 1,\ldots,n\}$, ν is updated by setting $\nu^{[c+2]}$ equal to the solution of the equation

$$-\phi\left(\frac{\nu}{2}\right)+\log\left(\frac{\nu}{2}\right)+\phi\left(\frac{\nu+p}{2}\right)-\log\left(\frac{\nu+p}{2}\right)$$
$$+\frac{1}{n}\sum_i\left[\log \tilde{w}_i^{[c+2]}-\tilde{w}_i^{[c+2]}\right]+1=0, \qquad (\mathbf{3.4.2})$$

where $\tilde{w}_i^{[c+2]} = (\nu+p)/(\nu+d_i^{[c+1]})$. Equations (3.4.1) and (3.4.2) are special cases of equations (27) and (30) given in Liu and Rubin (1995b).

Although using $Y_{\text{aug}} = \{y_i, i = 1,\ldots,n\}$ does not require an E-step (since Y_{aug} is fully observed), it results in a much more costly iteration for updating ν then $Y_{\text{aug}} = \{(y_i, q_i), i = 1,\ldots,n\}$ since every time (3.4.2) is evaluated during numerical optimization, the weights $\tilde{w}_i^{[c+2]}$, which depend on ν, must be recomputed and the function $\phi(\cdot)$ must be evaluated twice. Liu and Rubin's (1995b) simulation shows that, in their implementation, solving (3.4.2) for ν actually took about seven times longer than solving (3.4.1) for ν. The hope is that augmenting less will result in an algorithm that requires fewer iterations for convergence, which will make up for the extra computational burden per iteration. This was certainly the case in the examples presented in Liu and Rubin (1995a and 1995b), but as we shall see shortly, is not the case in our simulations.

In order to compare the performance of the four algorithms described in

Table 3.2, we fit a ten-dimensional t-model to 100 observations generated from $t_{10}(0, V, \nu = 1)$, where V was randomly selected at the outset of the simulation as a positive definite, non-diagonal matrix. The half-interval method (e.g., Carnahan, Luther, and Wilkes, 1969) was used to numerically update ν. Using the same convergence criterion and starting values for (μ, Σ) as described in the simulation of Section 2.3 and the starting value $\nu^{(0)} = 10$, the number of iterations required for convergence was recorded for each of the four algorithms. Figure 3.1 contains scatter plots which compare AECM 1 to each of the other three algorithms. As we see, AECM 1 was 8 – 12 times faster than either MCECM or ECME. Remember that the cost per iteration is less for AECM 1 and MCECM than for ECME. Moreover, AECM 2 (the combination of the optimal EM algorithm and ECME) was only slightly more efficient than AECM 1 in terms of the number of iterations required, and less efficient in terms of actual computer time.

In our simulation the choice of data augmentation when updating ν made little difference in terms of the number of iterations required for convergence. Yet Liu and Rubin (1995a and 1995b) have shown that ECME can be much more efficient than MCECM. There are two principal differences between our simulations and their examples. Specifically, both of their examples (example 3.1 of Liu and Rubin 1995a and "artificial example" of Liu and Rubin, 1995b) are two dimensional (as opposed to ten dimensional in our simulation) and contain much missing data among the y_i, whereas y_i was completely observed in our simulations. Figure 3.2 contains the results of a replication of our simulations on two-dimensional data and again indicates that ECME offers little advantage over MCECM (although the

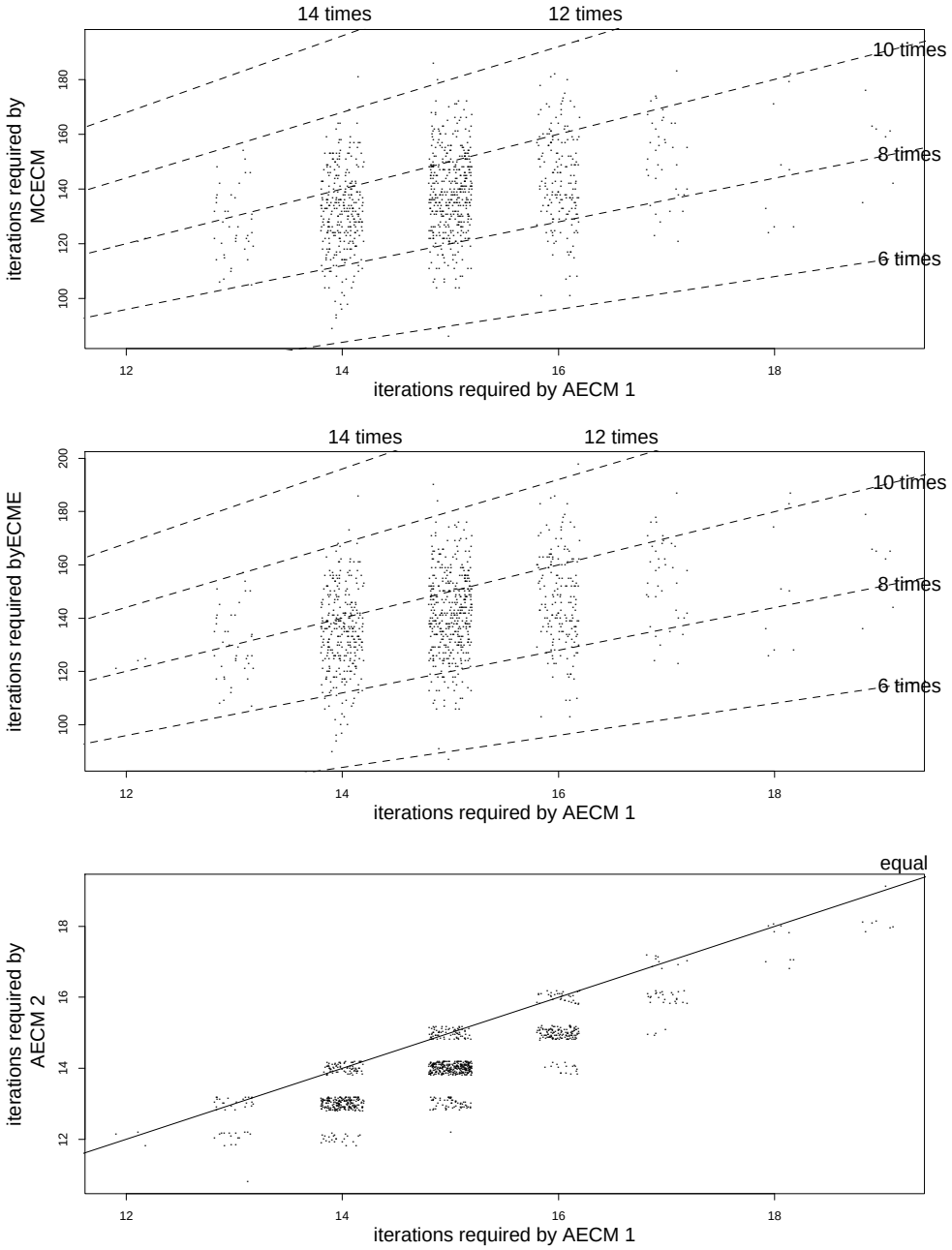

Figure 3.1. Comparing AECM algorithms for fitting the multivariate t with unknown degrees of freedom. The scatter plots compare the number of iterations required for convergence by each of MCECM, ECME, and AECM 2 with AECM 1 respectively. AECM 1 and 2 use the optimal data-augmentation of Section 2.3 for (μ, Σ) and perform very well. ECME and AECM 2 use Y_{obs} in place of Y_{aug} when updating ν and show only a small improvement over MCECM and AECM 1 respectively. This improvement is washed out by the increased computation time required per iteration by AECM 2

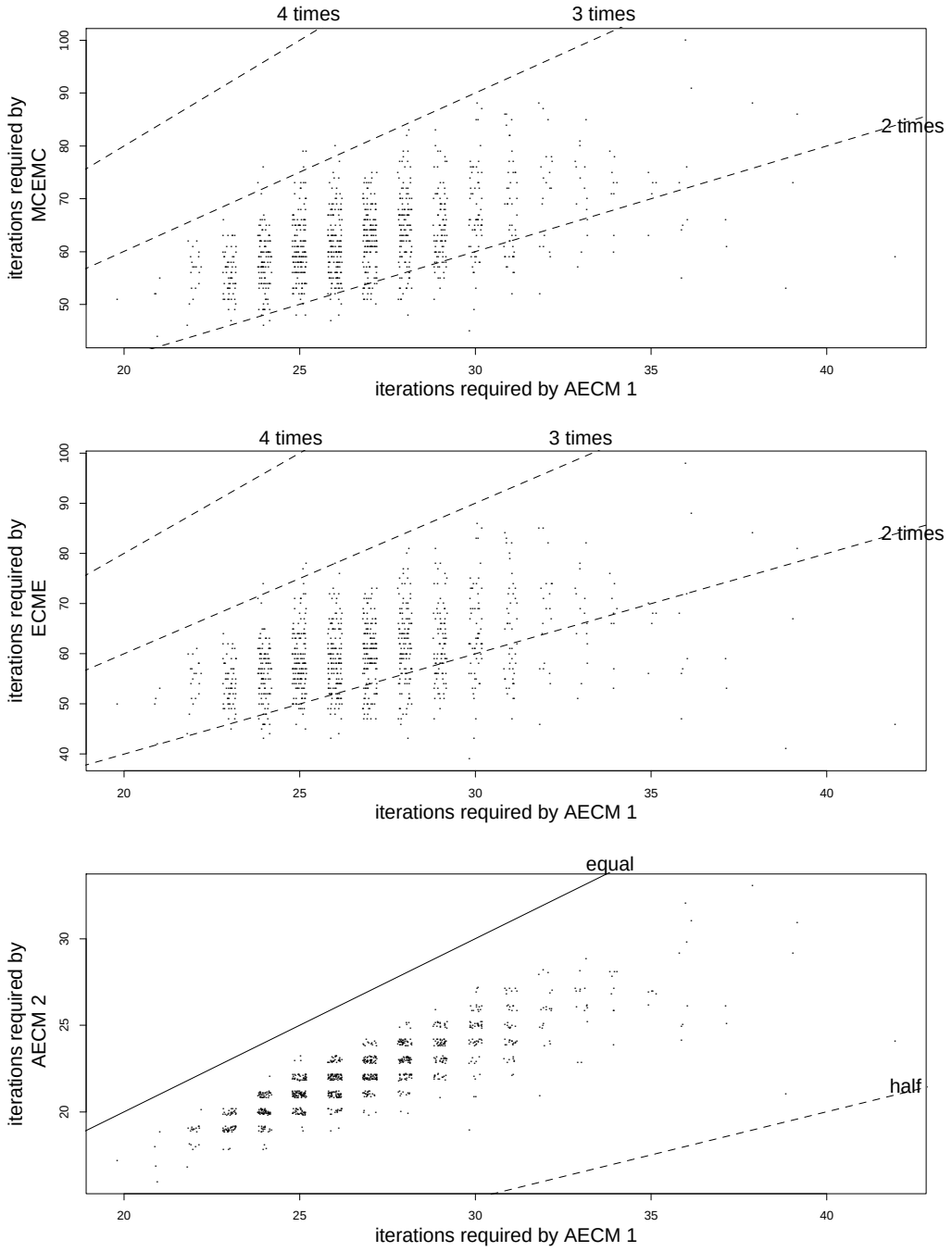

Figure 3.2. Comparing AECM algorithms for fitting the bivariate t with unknown degrees of freedom. The scatter plots compare the number of iterations required for convergence by each of MCECM, ECME, and AECM 2 with AECM 1 respectively. The computational gain of not augmenting when updating ν is again dwarfed by the gain of using the optimal augmentation for (μ, Σ).

improvement is somewhat greater than in the 10 dimensional case). Thus, it seems that when fitting t-models, ECME is most useful when there is much missing data among the y_i. How this interacts with the optimal augmentation for (μ, Σ) has yet to be investigated.

Chapter 4

The SECM Algorithm: Computing the Asymptotic Variance

4.1. Introduction

In the previous chapter we discussed how the AECM algorithm can be used to compute maximum likelihood estimates in the presence of missing data. Generally, statistical inference requires not only point estimates but also measures of uncertainty, for example (asymptotic) variance-covariance matrix of the estimates. The Supplemented EM (SEM) algorithm (Meng and Rubin, 1991a) computes such matrices using a sequence of EM iterates to obtain the matrix rate of convergence of EM. This rate is then used to inflate the augmented-data asymptotic variance-covariance matrix to obtain the asymptotic variance matrix for the observed-data MLEs. A key feature of SEM is that it requires only the code for EM and the code for computing the augmented-data asymptotic variance-covariance matrix.

In this chapter we develop and illustrate an analogous supplemented algorithm for ECM, SECM, which computes the asymptotic variance-covariance matrix

of the MLEs. In addition to requiring the computation of both the rate of convergence of ECM and the augmented-data variance-covariance matrix, it requires the computation of the rate of convergence of the CM algorithm. The computations of SECM, however, remain as simple as SEM in the sense that they only require the ECM code along with the code for computing the augmented-data variance-covariance matrix. Although our presentation is focused on the asymptotic variance-covariance matrix of the MLEs, the SECM algorithm can just as easily be applied to compute the asymptotic posterior variance-covariance matrix when ECM is used to find a posterior mode, which includes penalized likelihood models as a special case (e.g., Segal, Bacchetti, and Jewell, 1994).

After reviewing the SEM algorithm and other necessary theoretical development in Section 4.2, in Section 4.3 we detail the computational steps of SECM. Section 4.4 presents four examples to illustrate a variety of applications of the SECM algorithm. Section 4.5 offers discussion on some practical issues involved in implementing SECM, in particular, we will discuss the computation of asymptotic variance-covariance matrix when other EM-type algorithms are implemented.

4.2. Methodological Development

4.2.1. The SEM algorithm

The EM algorithm, as described in Section 1.2 is designed to calculate $\theta^\star$ the maximizer of the observed-data loglikelihood $L(\theta|Y_{\text{obs}}) = \log f(Y_{\text{obs}}|\theta)$. The SEM algo-

rithm supplements the EM algorithm in order to calculate the asymptotic variance-covariance matrix of $(\theta - \theta^\star)$, that is, $I_{\rm obs}^{-1}$, which is defined in (1.6.5). The algorithm is based on the fundamental identity, $DM^{EM} = I_{\rm mis} I_{\rm aug}^{-1}$, as presented in (1.6.2). This identity underlies the SEM computations because the desired information matrix, $I_{\rm obs}$, can be written as the difference between the augmented and missing information (e.g., Orchard and Woodbury, 1972; Meng and Rubin, 1991a)

$$I_{\rm obs} = I_{\rm aug} - I_{\rm mis} = [I_d - I_{\rm mis} I_{\rm aug}^{-1}] I_{\rm aug} = [I_d - DM^{EM}] I_{\rm aug}, \qquad (\textbf{4.2.1})$$

where I_d is the $d \times d$ identity matrix. In other words, to compute $I_{\rm obs}$, we need only compute DM^{EM} and $I_{\rm aug}$. When the augmented-data model, $f(Y_{\rm aug}|\theta)$ is from the exponential family, as is typically the case when the E-step is computable, $I_{\rm aug} = I_o(\theta^\star | S^\star(Y_{\rm obs}))$, where $S^\star(Y_{\rm obs}) = \mathsf{E}(S(Y_{\rm aug})|Y_{\rm obs}, \theta^\star)$, as found at the last E-step; we thus can compute $I_{\rm aug}$ using standard augmented-data procedures. Computing DM^{EM} can be accomplished by numerical differentiation of the EM mapping. Details of these SEM calculations are provided in Meng and Rubin (1991a) and will be reviewed in Section 4.3 .

4.2.2. The matrix rate of convergence of CM and ECM

To apply the logic of the SEM procedure to ECM, we need to relate the matrix rate of convergence of ECM to the fraction of missing information. Meng (1994) extended (1.6.2) to the ECM case with the following result:

$$[I_d - DM^{ECM}] = [I_d - DM^{EM}][I_d - DM^{CM}], \qquad (\textbf{4.2.2})$$

where DM^{ECM} is the rate of convergence of ECM at $\theta = \theta^\star$, and DM^{CM} is

the rate of convergence of CM at $\theta = \theta^\star$. If we knew DM^{ECM} and DM^{CM}, we could use (4.2.2) and (4.2.1) to calculate the asymptotic variance, $V_{\text{obs}} \equiv I_{\text{obs}}^{-1}$.

As will be detailed in Section 4.3, DM^{ECM} can be computed by numerical differentiation just like DM^{EM}. The rate of CM, DM^{CM}, can be computed in two ways. Meng (1994) shows that it can be calculated analytically as given in (1.6.7) and (1.6.8). All the quantities in (1.6.7) involve only the augmented-data information matrix and the g functions, and thus, they can be computed once $\theta^\star$ is obtained.

Alternatively, when the augmented-data model $f(Y_{\text{aug}}|\theta)$ is from an exponential family, DM^{CM} can be obtained numerically from the output of ECM at convergence, $(\theta^\star, S^\star(Y_{\text{obs}}))$, and an additional run of the code for the CM-steps. If we take $S^\star(Y_{\text{obs}})$ to be the fixed augmented-data sufficient statistics, we can obtain $\hat{\theta}(S^\star(Y_{\text{obs}}))$, the MLE of θ given $S^\star(Y_{\text{obs}})$, using the CM algorithm starting from $\theta^{(0)} \neq \theta^\star$; DM^{CM} is the rate of convergence of this CM algorithm. This can be proved by examining (1.6.8) and noting that if $f(Y_{\text{aug}}|\theta)$ is from an exponential family, $I_{\text{aug}} = I_o(\theta^\star|S^\star(Y_{\text{obs}}))$ and that $\theta^\star = \hat{\theta}(S^\star(Y_{\text{obs}}))$. Thus, we can derive DM^{CM} by calculating the rate of convergence of the CM algorithm applied to $L(\theta|S^\star(Y_{\text{obs}}))$. This avoids the matrix inversions and computation of the ∇_s in (1.6.7) and (1.6.8), which are necessary when performing analytical calculations.

With the PECM algorithm described in Section 1.3, the computation of DM^{CM} is particularly easy. Let Υ be the block diagonal matrix of I_{aug} with S blocks corresponding to the S subvectors of θ defined by the partition. Let Γ be the corresponding lower block triangular matrix of I_{aug}, that is, $I_{\text{aug}} = \Upsilon + \Gamma + \Gamma^\top$.

Meng (1990) established that in this case (1.6.7) reduces to

$$DM^{CM} = -\Gamma[\Upsilon + \Gamma^{\top}]^{-1}, \tag{4.2.3}$$

which makes analytical calculation of DM^{CM} very simple, as illustrated in Section 4.4.2.

4.2.3. The basic identity for the SECM algorithm

Having obtained DM^{ECM}, DM^{CM}, and I_{aug}, we can combine (4.2.1) and (4.2.2) to obtain

$$I_{\text{obs}} = I_o(\theta^{\star}|Y_{\text{obs}}) = [I_d - DM^{ECM}][I_d - DM^{CM}]^{-1} I_{\text{aug}}. \tag{4.2.4}$$

Equivalently, in terms of the variance,

$$V_{\text{obs}} \equiv I_{\text{obs}}^{-1} = V_{\text{aug}} + \Delta V, \tag{4.2.5}$$

where $V_{\text{aug}} = I_{\text{aug}}^{-1}$ can be viewed as the variance-covariance matrix of the MLE given the augmented data, and

$$\Delta V = V_{\text{aug}}[DM^{ECM} - DM^{CM}][I_d - DM^{ECM}]^{-1} \tag{4.2.6}$$

is the increase in variance due to the missing data.

Identity (4.2.6) is the basis for the SECM algorithm, and it reduces to (2.3.6) of Meng and Rubin (1991a) when $DM^{CM} = 0$, which corresponds to EM. An interesting property of SECM (and SEM) is that, although ΔV is mathematically symmetric, the right side of (4.2.6) is not numerically constrained to be symmetric because V_{aug}, DM^{CM}, and DM^{ECM} are computed separately as described in

Section 4.3. Numerical symmetry is obtained only when all three of these are computed without appreciable numerical imprecision. This property turns out to be a surprisingly powerful tool for detecting implementation or numerical errors, as illustrated in Section 4.4 and further discussed in Section 4.5.

4.2.4. When some components have no missing information

In certain situations, missing data only affect estimates for some components of θ, that is, there is no missing information for the rest of θ. For example, with $\theta = (\vartheta_1, \vartheta_2)$, there might be no missing data for the estimate of ϑ_1, in which case we can compute the MLE $\vartheta_1^\star$ without using EM or ECM (see Section 4.4.5). An efficient implementation of ECM in such cases fixes $\vartheta_1^{(t)} = \vartheta_1^\star$ and only updates $\vartheta_2^{(t)}$. For comparison with ECM applied to θ, for notational convenience, we will refer to this version of the ECM algorithm as ECM*.

Since ECM* is really a special case of ECM, the corresponding SECM algorithm (we will call it SECM*) can be used to calculate $\Delta V(\vartheta_2^\star|\vartheta_1^\star)$, the increase in asymptotic conditional variance of $\vartheta_2^\star$ (conditioning on $\vartheta_1^\star$) due to missing information. Specifically, we can compute $\Delta V(\vartheta_2^\star|\vartheta_1^\star)$ (see (4.2.6)) as

$$\Delta V(\vartheta_2^\star|\vartheta_1^\star) = \{[I_{\text{aug}}]_{22}\}^{-1}\,[DM^{ECM^\star} - DM^{CM^\star}][I_{d_2} - DM^{ECM^\star}]^{-1}, \quad \textbf{(4.2.7)}$$

where the CM* algorithm is run with $\vartheta_1^{(t)}$ fixed at $\vartheta_1^\star$, $[I_{\text{aug}}]_{22}$ is the submatrix of I_{obs} corresponding to ϑ_2, and d_2 is the dimension of ϑ_2. It turns out that (4.2.7) is all we need to compute the increase in variance, since

$$\Delta V = \begin{pmatrix} 0 & 0 \\ 0 & \Delta V(\vartheta_2^\star|\vartheta_1^\star) \end{pmatrix}. \quad \textbf{(4.2.8)}$$

This identity holds because (i) there is no increase in variance or covariance of $\vartheta_1^\star$ and (ii) there is no increase in the part of the variance of $\vartheta_2^\star$ that can be explained by $\vartheta_1^\star$ (see Meng and Rubin, 1991a). When there is no missing information for ϑ_1, we can therefore calculate ΔV using (4.2.8) and then calculate V_{obs} using (4.2.5). It is, however, worth remarking that fixing $\vartheta_1^{(t)}$ at $\vartheta_1^\star$ increases the efficiency of ECM and SECM but is not a required step since the standard ECM and SECM algorithms will produce the desired estimates. This is in contrast to the standard SEM algorithm, which fails in this situation and a special version of SEM *must* be implemented. An example illustrating these points is given in Section 4.4.5.

4.3. Implementing the SECM Algorithm

4.3.1. A schematic

This section is designed to explain how to implement SECM in a step-by-step manner. Readers not interested in details of implementation may wish to skip to the examples in Section 4.4. We will describe in simple terms exactly how one can compute $\theta^\star$ and V_{obs}. The schematic in Figure 4.1 describes the necessary steps in broad terms. The user must provide routines that perform the E- and CM-steps, as well as one that computes I_{aug}. These are described in Section 4.3.2. The schematic also references `Algorithms 1, 2,` and `3,` which compute $\theta^\star$, DM^{ECM}, and DM^{CM} respectively and are described in Section 4.3.3. The mathematical background for the routines that follow is given in Section 4.2 of this paper and

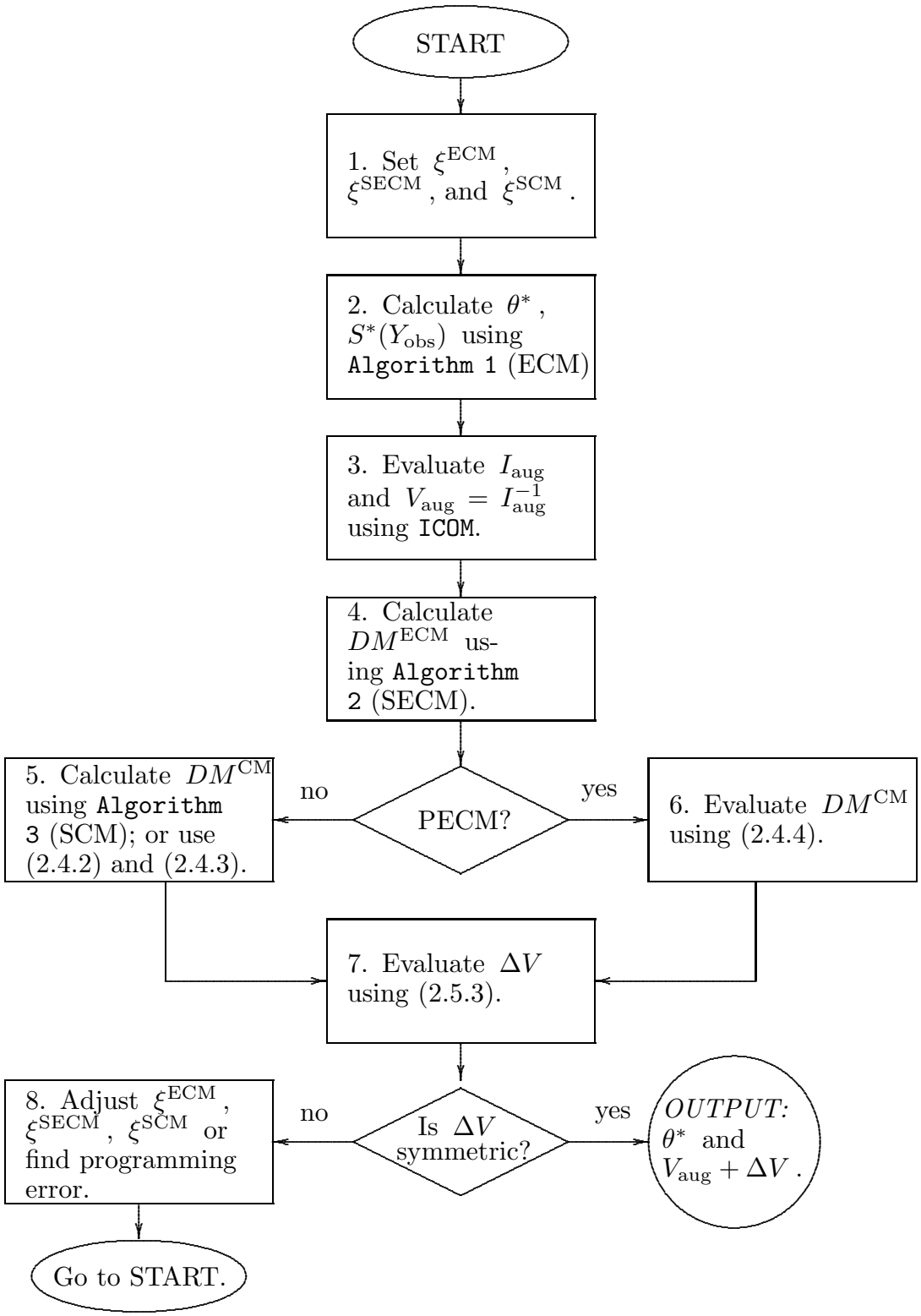

Figure 4.1. Schematic of the SECM algorithm.

in Meng and Rubin's (1991a) presentation of the SEM algorithm. Since SEM is a special case of SECM, the algorithms presented here can also be used to implement SEM. The only modification when running SEM is that the `CMSTEPS` routine should compute the global maximum of $Q(\theta|\theta^{(t)})$, that is, use only one CM-step, and the DM^{CM} matrix should be set to 0.

4.3.2. User provided specific subroutines

The computations in the following three subroutines are problem specific; the first two routines are used in the Algorithms in Section 4.3.3 and the third is used in Box 3 of Figure 4.1. These subroutines are developed assuming that $f(Y_{\text{aug}}|\theta)$ is from an exponential family, beyond which the simplicity of EM-type algorithms is typically lost because the E-step typically requires numerical integration or simulations (c.f. Wei and Tanner, 1990).

`Subroutine 1. ESTEP:`

INPUT: $\theta^{(t)}$, Y_{obs}

Compute $S^{(t)}(Y_{\text{obs}}) = \mathsf{E}[S(Y_{\text{aug}})|\theta^{(t)}, Y_{\text{obs}}]$, where $S(Y_{\text{aug}})$ is the augmented-data sufficient statistic.

OUTPUT: $S^{(t)}(Y_{\text{obs}})$

`Subroutine 2. CMSTEPS:`

INPUT: $S^{(t)}(Y_{\text{obs}})$, $\theta^{(t)}$

Compute $\theta^{(t+1)}$ with a sequence of constrained maximization steps, as described in Section 1.3.

OUTPUT: $\theta^{(t+1)}$

`Subroutine 3. ICOM:`

INPUT: $\theta^\star$, $S^\star(Y_{\text{obs}})$

Compute $I_{\text{aug}} = I_o(\theta^\star|S^\star(Y_{\text{obs}}))$, the observed Fisher information matrix based on the augmented-data model, evaluated at $\theta^\star$ and $S^\star(Y_{\text{obs}})$.

OUTPUT: I_{aug} and $V_{\text{aug}} = I_{\text{aug}}^{-1}$

4.3.3. General algorithms

`Algorithm 1:` Calculate $\theta^\star$ and $S^\star(Y_{\text{obs}})$ using ECM.
Repeat the ECM steps:

INPUT: $\theta^{(t)}$

Step 1: Calculate $S^{(t)}(Y_{\text{obs}})$ with `ESTEP`;

Step 2: Calculate $\theta^{(t+1)}$ with `CMSTEPS`;

OUTPUT: $\theta^{(t+1)}$
Continue until

$$\delta(\theta^{(t)}, \theta^{(t+1)}) < \xi^{ECM} \qquad (\textbf{4.3.1})$$

for some convergence criteria δ and threshold ξ^{ECM} . A discussion on how to choose δ and ξ^{ECM} , as well as ξ^{SECM} and ξ^{SCM} , appears in Section 4.3.4.

FINAL OUTPUT : Set $S^\star(Y_{\text{obs}})$ equal to the output from the last `ESTEP`, and set $\theta^\star$ equal to the output from the last `CMSTEPS`.

`Algorithm 2:` Calculate DM^{ECM} using SECM.

Let r_{ij} be the (i,j) th element of the $d \times d$ matrix DM^{ECM} and define $\theta^{(t)}(i)$ as

$$\theta^{(t)}(i) = (\theta_1^\star, \ldots, \theta_{i-1}^\star, \theta_i^{(t)}, \theta_{i+1}^\star, \ldots, \theta_d^\star), \qquad i = 1, \ldots, d. \tag{4.3.2}$$

That is, $\theta^{(t)}(i)$ is $\theta^\star$ with the i th component active, i.e. replaced by the i th component of $\theta^{(t)}$.

Repeat the SECM steps:

INPUT: $\theta^\star$ and $\theta^{(t)}$

Repeat steps 1 and 2 for each i

Step 1: Calculate $\theta^{(t)}(i)$ from (4.3.2), treat it as input for `Algorithm 1`, and run one iteration of `Algorithm 1` (that is, one `ESTEP` and one `CMSTEPS`) to obtain $\tilde{\theta}^{(t+1)}(i)$;

Step 2: Obtain the ratio

$$r_{ij}^{(t)} = \frac{\tilde{\theta}_j^{(t+1)}(i) - \theta_j^\star}{\theta_i^{(t)} - \theta_i^\star} \quad \text{for} \quad j = 1, \ldots, d; \tag{4.3.3}$$

OUTPUT: $\{r_{ij}^{(t)}, i, j = 1, \ldots, d\}$.

FINAL OUTPUT: $DM^{ECM} = \{r_{ij}^\star\}$, where $r_{ij}^\star = r_{ij}^{(t_{ij})}$ is such that

$$\delta(r_{ij}^{(t_{ij})}, r_{ij}^{(t_{ij}+1)}) < \xi^{SECM} \tag{4.3.4}$$

for some suitable convergence threshold ξ^{SECM}.

`Algorithm 3`: Calculate DM^{CM} using SCM.

For notational simplicity, the same notation is used for the elements of DM^{CM} as was used for those of DM^{ECM}.

Repeat the SCM (e.g. supplemented CM) steps:

INPUT: $\theta^{\star}$, $\theta^{(t)}$, and $S^{\star}(Y_{\text{obs}})$

Repeat steps 1 and 2 for each i

Step 1: Calculate $\theta^{(t)}(i)$ from (4.3.2) and run `CMSTEPS` once using $S^{\star}(Y_{\text{obs}})$ and $\theta^{(t)}(i)$ as input to obtain $\tilde{\theta}^{(t+1)}(i)$;

Step 2: Obtain the ratio $r_{ij}^{(t)}$ as in (4.3.3);

OUTPUT: $\{r_{ij}^{(t)}, i, j = 1, \ldots, d\}$.

FINAL OUTPUT: $DM^{CM} = \{r_{ij}^{\star}\}$ where all the $r_{ij}^{\star} = r_{ij}^{t_{ij}}$ are such that (4.3.4) is satisfied for some convergence threshold for SCM, ξ^{SCM}.

When implementing the PECM algorithm, `Algorithm 3` can be replaced by a simple evaluation of (4.2.3). For the more general ECM algorithm, it can be computationally advantageous to replace `Algorithm 3` with the analytical calculations described in (1.6.7) and (1.6.8). Finally, the outputs of `ICOM` and `Algorithms 2` and `3` (i.e. V_{aug}, DM^{ECM}, DM^{CM}) are put together to calculate ΔV using (4.2.6), and then (4.2.5) is used to obtain the desired variance-covariance matrix V_{obs}.

4.3.4. Notes on implementation

The convergence criterion $\delta(a, b)$ is a discrepancy measure between a and b. Common choices are (i) $\delta(a, b) = \max_i |a_i - b_i|$, (ii) $\delta(a, b) = \max_i |(a_i - b_i)/a_i|$

or $\max_i |(a_i - b_i)/(a_i + b_i)|$, and (iii) $\delta(a,b) = ||a - b||$, where $||\cdot||$ denotes the standard Euclidean norm, and a_i and b_i are the components of a and b. The first of these, (i), was used in the examples in Section 4.4 and is generally fine unless the magnitudes of the components vary greatly, in which case (ii) is preferred. The same holds for SECM and SCM except that a and b are scalars, in which case (i) and (iii) are the same.

When ECM is run alone, the convergence threshold ξ^{ECM} can be set to obtain whatever level of precision is desired for $\theta^\star$. When SECM is used, however, ξ^{ECM} must be quite small (compared to the magnitude of $\theta^\star$) to insure convergence of $\theta^{(t)}$ as well as $r_{ij}^{(t)}$ and thereby to insure satisfactory symmetry in ΔV. Generally, ξ^{SECM} and ξ^{SCM} should be about equal to the square root of ξ^{ECM}, as discussed further in Section 4.5.1. Finally, note that `Algorithms 2` and `3` assume that the ECM iterates were saved when `Algorithm 1` was run. This saves computational time, but requires extra storage. For some users, it may be better to recompute the iterates than to save them. To do this, change the input in `Algorithms 2` and `3` to "*INPUT:* $\theta^\star$ and $\theta^{(t-1)}$" and add "*Step 0:* Run one ECM iteration on $\theta^{(t-1)}$ to obtain $\theta^{(t)}$." Generally, it is not necessary to start the SECM or SCM algorithms at $\theta^{(0)}$ or to run them for as many steps as ECM was run. Thus, saving all the iterates may not be economical, and it may be computationally more efficient to recompute only the iterates that are needed.

4.4. Examples Illustrating the SECM Algorithm

4.4.1. Introduction to examples

In this section we present several examples that illustrate different aspects of SECM. The first example is a simple bivariate normal used in Meng (1994) to illustrate several surprising phenomena concerning the rates of convergence of EM, ECM, and their variations. It shows how the basic building blocks of ECM are put together. We also compare our simulation results to the theoretical results presented in Meng (1994).

Our second example is a bivariate normal with stochastic censoring, often applied in economics and known as the Type II Tobit model (e.g. Amemiya, 1984). We demonstrate the easy implementation of SECM for this class of practically useful models. We also show how inspecting the symmetry of the resulting variance-covariance matrix led to the discovery of an error in the literature for computing the E-step.

The third example illustrates a class of important applications of SECM – computing the asymptotic variance-covariance matrix of maximum likelihood estimates fit to contingency tables in the presence of incomplete observations. The algorithm essentially uses only the standard Iterative Proportional Fitting, or IPF algorithm (e.g., Bishop, Feinberg and Holland, 1975) and a simple E-step that allocates the counts according to the estimated conditional cell probabilities.

Our last example is presented for numerical comparison with SEM results. Meng and Rubin (1991a) uses this example to illustrate a variety of issues involved

in implementing EM and SEM. We repeat these for ECM and SECM, which allows us to make direct comparisons with the results in Meng and Rubin (1991a).

4.4.2. A simple theoretical example

Suppose the augmented data consist of two observations from a bivariate normal distribution

$$Y_i = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} \overset{\text{iid}}{\sim} N\left[\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right], \qquad i = 1, 2,$$

but we only observe $z_1 = y_{11}$ and $z_2 = y_{22} - y_{21}$. Here ρ is known and we are interested in the MLE $\theta^\star$ of $\theta = (\theta_1, \theta_2)$ based on $z_1 \sim N(\theta, 1)$ and $z_2 \sim N(\theta_2 - \theta_1, 2(1-\rho))$. In this case the MLE is in closed form, $\theta^\star = (z_1, z_1 + z_2)$, with variance-covariance matrix $\begin{pmatrix} 1 & 1 \\ 1 & 3-2\rho \end{pmatrix}$ since z_1 and z_2 are independent. Thus, this example allows us to compare empirical results with the theoretical values.

The E- and CM-steps follow simply from properties of the bivariate normal distribution. Since ρ is known, the augmented-data sufficient statistics are simply $(\bar{y}_1, \bar{y}_2)$, where $\bar{y}_j = \frac{1}{2}(y_{1j} + y_{2j})$. Thus,

$$\text{E}: \quad \begin{cases} \bar{y}_1^{(t)} = \mathsf{E}(\bar{y}_1 | z_1, z_2, \theta^{(t)}) = \frac{z_1}{2} + \frac{1}{4}(\theta_1^{(t)} + \theta_2^{(t)} - z_2) \\ \bar{y}_2^{(t)} = \mathsf{E}(\bar{y}_2 | z_1, z_2, \theta^{(t)}) = \frac{1}{2}(\theta_2^{(t)} + \rho(z_1 - \theta_1^{(t)})) + \frac{1}{4}(\theta_1^{(t)} + \theta_2^{(t)} + z_2), \end{cases}$$

$$\text{CM}_1: \quad \theta_1^{(t+\frac{1}{2})} = \bar{y}_1^{(t)} + \rho(\theta_2^{(t)} - \bar{y}_2^{(t)}),$$

$$\text{CM}_2: \quad \theta_2^{(t+\frac{2}{2})} = \bar{y}_2^{(t)} + \rho(\theta_1^{(t+\frac{1}{2})} - \bar{y}_1^{(t)}),$$

Notice that the E-step can easily be derived using the fact that $y_{21} + y_{22}$ is independent of $y_{22} - y_{21}$ and that the latter is observed. Iterating these three steps as in Algorithm 1 leads to $\theta^\star$. To implement SECM, we first run Algorithm 2 of

Section 4.3 to obtain DM^{ECM} (which depends on the known value of ρ). In this case, DM^{CM} is particularly easy to compute because this is an example of PECM. It is easy verify that

$$I_{\text{aug}} = \frac{2}{1-\rho^2}\begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \qquad \text{and} \qquad V_{\text{aug}} = I_{\text{aug}}^{-1} = \frac{1}{2}\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Thus, (4.2.3) gives

$$DM^{CM} = -\Gamma[\Upsilon + \Gamma^{\top}]^{-1} = \begin{pmatrix} 0 & 0 \\ \rho & \rho^2 \end{pmatrix}.$$

Once we have DM^{ECM}, DM^{CM} and V_{aug}, the desired matrix V_{obs} follows immediately from (4.2.5) and (4.2.6).

We ran two simulations, one with $\rho = -0.5$ and one with $\rho = 0.5$. For both simulations we used $\xi^{ECM} = 10^{-8}$ and $\xi^{SECM} = 10^{-6}$. The calculated values of $\theta^{\star}$, DM^{ECM}, and V_{obs} all agree with the theoretical results given above to 6 decimal places. The simulated data and results appear in Table 4.1.

4.4.3. Bivariate normal stochastic censoring model

Suppose $(y_{i1}, y_{i2})^{\top}$ are independent observations from a bivariate normal distribution, where y_2 is never observed and y_1 is observed only if $y_2 > 0$. For each unit, the density is specified by

$$\begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} \overset{\text{indep}}{\sim} N\left[\begin{pmatrix} \beta_{11}x_{i1} + \beta_{12}x_{i2} + 0 \cdot x_{i3} \\ \beta_{21}x_{i1} + 0 \cdot x_{i2} + \beta_{23}x_{i3} \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1 \\ \rho\sigma_1 & 1 \end{pmatrix}\right], \quad i = 1, \ldots, n.$$

Here the x_{ij} are augmented observed, and we set $\beta = (\beta_{11}, \beta_{12}, \beta_{21}, \beta_{23})$ and $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1 \\ \rho\sigma_1 & 1 \end{pmatrix}$.

ρ	$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}$	$\begin{pmatrix}z_1\\ z_2\end{pmatrix}$	$\begin{pmatrix}\theta_1^\star\\ \theta_2^\star\end{pmatrix}$	V_{aug}	DM^{EM}	DM^{CM}	DM^{ECM}	V_{obs}
-0.5	5	4.9663	4.9663	.50 $-.25$	.25 .50	0 0	.500 .3750	1 1
	-10	-12.6183	-7.6520	$-.25$ .50	.25 .75	$-.5$ .25	.125 .8125	1 4
0.5	5	3.9832	3.9832	.50 .25	.25 0	0 0	.250 0	1 1
	-10	-15.9711	-11.9879	.25 .50	.25 .75	.5 .25	.375 .8125	1 2

Table 4.1. Results for a simple bivariate normal example (Section 4.4.2).

This is an example of the so-called seemingly unrelated regression model (Zellner, 1962), also known in economics as the Type II Tobit model (e.g., Amemiya, 1984). When the active covariates for y_{i1} and y_{i2} overlap but are not identical (in our example, $\beta_{13} = \beta_{22} = 0$), even if all the y's are observed, the MLEs of β and Σ are not in closed form. Consequently, implementing EM would require nested iterations within the M-step. However, given β, the conditional MLE of Σ is simply the sum of squares of the residuals divided by n. On the other hand, given Σ, the conditional MLE for β can be easily obtained by weighted least squares. ECM replaces the iterative M-step with these two CM-steps (detailed formulas are given in Meng and Rubin, 1994b).

To compute the E-step, we need to find the conditional expectation of $(y_{i1}, y_{i1}^2, y_{i2}, y_{i2}^2, y_{i1}y_{i2})$ for $i = 1, 2, \ldots, n$, given the observed data and the parameters. These calculations follow from the properties of the bivariate normal distribution and are given explicitly in Little and Rubin (1987, p. 225). There is, however, an error in that presentation. When $y_{i2} > 0$, we also observe y_{i1}, and must thus find $\mathsf{E}(y_{i2}|y_{i1}, y_{i2} > 0)$ and $\mathsf{E}(y_{i2}^2|y_{i1}, y_{i2} > 0)$, not $\mathsf{E}(y_{i2}|y_{i2} > 0)$ and $\mathsf{E}(y_{i2}^2|y_{i2} > 0)$ as presented there. This error will lead to incorrect results; in particular, it tends to underestimate the magnitude of ρ. For the data described below, the true $\rho = 0.5$, the MLE $\rho^\star = 0.482$, but the incorrect procedure gives 0.200. We discovered this error only after we found that the resulting variance-covariance matrix from SECM was clearly asymmetric, which demonstrates the power of SECM as a tool for detecting errors in implementing ECM. To correct the E-step, we need to substitute the following two expectations for the second and fifth equations given

x_1	x_2	x_3	Y_1	Y_2
−1	1	−1	−0.4443346	−2.9841022
−1	1	−1	−0.4038098	−0.9029128
−1	−1	−1	−0.4457312	−0.1776825
−1	−1	0	−0.1966688	0.4006104
0	1	0	0.5583971	0.3723503
0	−1	0	−0.7892194	1.1994856
0	1	2	−0.2868998	−0.5555625
0	−1	2	−0.4309087	0.7991658
1	1	2	1.2447119	1.4188357
1	1	3	1.3696260	2.1091285
1	−1	3	−0.4198308	0.0973109
1	−1	3	−0.3999554	1.1703623

Table 4.2. The data for the stochastic censoring example (Section 4.4.2).

in Little and Rubin (1987, p. 225):

$$\mathsf{E}(y_{i2}|y_{i1}, y_{i2} > 0, \beta^{(t)}, \Sigma^{(t)}) = \eta_{i2}^{(t)} + \tau_{i2}^{(t)} \lambda\left(\frac{\eta_{i2}^{(t)}}{\tau_{i2}^{(t)}}\right),$$

$$\mathsf{E}(y_{i2}^2|y_{i1}, y_{i2} > 0, \beta^{(t)}, \Sigma^{(t)}) = [\tau_{i2}^{(t)}]^2 + [\eta_{i2}^{(t)}]^2 + \tau_{i2}^{(t)} \eta_{i2}^{(t)} \lambda\left(\frac{\eta_{i2}^{(t)}}{\tau_{i2}^{(t)}}\right),$$

where $\lambda(z) = \phi(z)/\Phi(z)$ is the inverse Mill's ratio, and

$$\eta_{i2} = \mathsf{E}(y_{i2}|y_{i1}, \beta, \Sigma) = \mu_2 + \rho\frac{(y_{i1} - \mu_1)}{\sigma_1}, \qquad \tau_{i2} = \sqrt{\mathsf{Var}(y_{i2}|y_{i1}, \beta, \Sigma)} = \sqrt{1 - \rho^2}.$$

We ran SECM using the variance stabilizing transformations $(\log(\sigma_1), Z_\rho)$ in place of (σ_1, ρ), where $Z_\rho = 0.5 \log\{(1 + \rho)/(1 - \rho)\}$ is the Fisher Z transformation of ρ. This transformation is used not only to ensure better normality when invoking large sample approximations, but also to enhance the computational stability of SECM since $DM^{ECM}(\theta)$ is more nearly constant (as a matrix function of θ) near $\theta^\star$ when the loglikelihood is more nearly quadratic. The sample data set of size 12 in Table 4.2 was simulated using the parameters in the first row of Table 4.3. The observed data are the 8 observations of Y_1 for which the corresponding

	β_{11}	β_{12}	β_{21}	β_{23}	$\log(\sigma_1)$	Z_ρ
θ_{true}	0.2000	0.5000	−0.3000	0.3000	−0.6931	0.5493
$\theta^\star$	0.2643	0.6248	−0.5263	0.5274	−1.0857	0.5253
$\mathsf{Var}\,(\theta^\star)$	0.0229	0.0147	0.2609	0.1614	0.0629	0.4948
ΔV	0.0094	0.0052	0.1769	0.1381	0.0231	0.4131

Table 4.3. Results for the stochastic censoring example (Section 4.4.2). The table records the parameter values used to generate the data in Table 4.2, the MLE of the parameter, its asymptotic variance, and the change in variance due to missing data.

observation of Y_2 is positive. None of the values of Y_2 are observed. ECM was run with $\xi^{ECM} = 10^{-12}$, resulting in 249 iterations, from a starting value of all zeros. Table 4.3 contains the MLEs of β, ρ, and σ_1 in the second row and the corresponding asymptotic variances, which were found using SECM as described below, in the third row.

Since the augmented-data distribution is from a standard exponential family, I_{aug} is just the augmented-data information matrix evaluated at $\theta^\star$ and $S^\star(Y_{\text{obs}})$, which yields the matrix in Figure 4.2. Since this is a PECM algorithm, using (4.2.3) we can quickly calculate DM^{CM} from I_{aug}. In applying (4.2.3), Υ is the block diagonal matrix indicated in Figure 4.2, and the nonzero portion of Γ is the lower left 2×4 submatrix of V_{aug}. SECM was run with $\xi^{SECM} = 10^{-7}$, and V_{obs} is found by a simple application of (4.2.5) and appears in Figure 4.3, which indicates that the symmetry holds to at least 4 decimal places verifying the accuracy of the computations. Comparing V_{aug} and V_{obs}, we can also easily find the increase in variance due to missing data, as recorded in the fourth row of Table 4.3. (We have applied a Jacobian transformation for the variances of $\sigma_1^\star$ and $\rho^\star$.)

$$
V_{\text{aug}} = \begin{array}{c} \\ \beta_{11} \\ \beta_{12} \\ \beta_{21} \\ \beta_{23} \\ \\ \log(\sigma_1) \\ Z_\rho \end{array}
\begin{array}{c}
\begin{array}{ccccccc} \beta_{11} & \beta_{12} & \beta_{21} & \beta_{23} & & \log(\sigma_1) & Z_\rho \end{array} \\
\left(\begin{array}{ccccccc}
0.013512 & 0.000000 & 0.001210 & 0.007259 & \vdots & -0.002169 & -0.008234 \\
0.000000 & 0.009502 & 0.013557 & 0.000000 & \vdots & 0.000000 & 0.000000 \\
0.001210 & 0.013557 & 0.083980 & 0.003880 & \vdots & -0.000009 & -0.002012 \\
0.007259 & 0.000000 & 0.003880 & 0.023278 & \vdots & -0.000051 & -0.012073 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-0.002169 & 0.000000 & -0.000009 & -0.000051 & \vdots & 0.039834 & 0.016296 \\
-0.008234 & 0.000000 & -0.002012 & -0.012073 & \vdots & 0.016296 & 0.081700
\end{array}\right)
\end{array}
$$

Figure 4.2. The augmented-data variance-covariance matrix for the stochastic censoring example (Section 4.4.3). When implementing PECM, the block structure can be used to easily calculate DM^{CM}.

$$V_{\text{obs}} = \begin{array}{c} \\ \beta_{11} \\ \beta_{12} \\ \beta_{21} \\ \beta_{23} \\ \\ \log(\sigma_1) \\ Z_\rho \end{array} \begin{array}{c} \begin{array}{ccccccc} \beta_{11} & \beta_{12} & \beta_{21} & \beta_{23} & & \log(\sigma_1) & Z_\rho \end{array} \\ \left(\begin{array}{ccccccc} 0.022862 & -0.002858 & -0.001767 & 0.007392 & \vdots & -0.000422 & -0.004376 \\ -0.002858 & 0.014679 & 0.008360 & 0.002188 & \vdots & -0.002109 & -0.015259 \\ -0.001765 & 0.008360 & 0.260939 & -0.081708 & \vdots & 0.005719 & 0.115485 \\ 0.007390 & 0.002187 & -0.081704 & 0.161360 & \vdots & -0.003444 & -0.143232 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -0.000421 & -0.002109 & 0.005719 & -0.003445 & \vdots & 0.062933 & 0.029605 \\ -0.004369 & -0.015253 & 0.115469 & -0.143233 & \vdots & 0.029602 & 0.494789 \end{array} \right) \end{array}$$

Figure 4.3. The observed-data variance-covariance matrix for the stochastic censoring example (Section 4.4.3). The symmetry holds to at least 4 decimal places, indicating accurate computations.

Clinic (C)	Prenatal Care (P)	Survival (S): Died	Survived	
(a) Completely Classified Cases				
A	Less	3	176	
	More	4	293	
B	Less	17	197	
	More	2	23	$N^{(a)} = 715$ cases
(b) Partially Classified Cases				
?	Less	10	150	
	More	5	90	$N^{(b)} = 255$ cases

Table 4.4. A $2 \times 2 \times 2$ contingency table with partially classified observations.

4.4.4. A $2 \times 2 \times 2$ contingency table

Table 4.4(a) presents a $2 \times 2 \times 2$ contingency table on infant survival (Bishop, Feinberg, and Holland, 1975, Table 2.4-2). The supplementary data in Table 4.4(b) was added by Little and Rubin (1987, p. 187) to form a partially classified table. Suppose we wish to fit a log-linear model with no three-way interaction:

$$\log(\theta_{ijk}) = u_0 + (-1)^{i-1}u_P + (-1)^{j-1}u_C + (-1)^{k-1}u_S + (-1)^{i+j}u_{PC} + (-1)^{j+k}u_{CS} + (-1)^{i+k}u_{PS}, \qquad (\mathbf{4.4.1})$$

where θ_{ijk} is the cell probability for cell (i, j, k) for $i, j, k = 1, 2$, with i corresponding to P, j to C and k to S. We will derive the MLE of $U = (u_0, u_P, \ldots, u_{PS})^\top$ and V_{obs}, the asymptotic variance-covariance matrix of $U^\star$.

Meng and Rubin (1991b, 1993) describe an ECM algorithm with three CM-steps for this problem. Specifically, since the loglikelihood is linear in the cell counts,

$Y = \{y_{ijk}\}$, the E-step simply involves imputing the missing data,

$$\mathrm{E} : y_{ijk}^{(t)} = \tilde{y}_{ijk}^{(a)} + \tilde{y}_{ik}^{(b)} \frac{\theta_{ijk}^{(t)}}{\sum_j \theta_{ijk}^{(t)}}, \qquad (\mathbf{4.4.2})$$

where $\tilde{y}_{ijk}^{(a)}$ are the cell counts in Table 4.4(a), and $\tilde{y}_{ik}^{(b)}$ are the marginal counts classified only according to parental care and survival (see Table 3(b)). The CM-steps make use of IPF. Given the current estimated cell probabilities $\{\theta_{ijk}^{(t)}\}$, the three CM-steps are

$$\mathrm{CM}_1 : \theta_{ijk}^{(t+\frac{1}{3})} = \theta_{ij(k)}^{(t)} \frac{y_{ij+}}{N}$$

$$\mathrm{CM}_2 : \theta_{ijk}^{(t+\frac{2}{3})} = \theta_{i(j)k}^{(t+\frac{1}{3})} \frac{y_{i+k}}{N}$$

$$\mathrm{CM}_3 : \theta_{ijk}^{(t+\frac{3}{3})} = \theta_{(i)jk}^{(t+\frac{2}{3})} \frac{y_{+jk}}{N}$$

where N is the total count, $\theta_{ij(k)} = \theta_{ijk} / \sum_k \theta_{ijk}$ is the conditional probability of the third factor given the first two, and $y_{ij+} = \sum_k y_{ijk}$, etc. It is easy to see that CM_1 maximizes $L(\theta|Y)$ subject to $\theta_{ij(1)} = \theta_{ij(1)}^{(t)}$ for each i and j, so that the constraint function $g_1(\theta) = \{\theta_{ij(1)}\}$. Likewise $g_2(\theta) = \{\theta_{i(1)k}\}$ and $g_3(\theta) = \{\theta_{(1)jk}\}$. It is clear that this is not a PECM algorithm.

We start ECM with $\theta_{ijk} = \frac{1}{8}$ for each i, j, and k, which satisfies the constraint of no three-way interaction, and cycle according to $\mathrm{ECM} : \mathrm{E} \to \mathrm{CM}_1 \to \mathrm{CM}_2 \to \mathrm{CM}_3$. At each iteration, $U^{(t)}$ can then be calculated by regression

$$U^{(t)} = (X^\top X)^{-1} X^\top \log \theta^{(t)}, \quad t = 1, 2, 3, \ldots \qquad (\mathbf{4.4.3})$$

where the design matrix X is derived from (4.4.1) with elements either $+1$ or -1. Notice that we are defining two mappings, $M_\theta : \theta^{(t)} \to \theta^{(t+1)}$ and $M_U : U^{(t)} \to U^{(t+1)}$. The θ parameterization is more natural in the context of the E

and CM-steps. The U parameterization is more convenient in the context of the log-linear model, and is a stable parameterization for the SECM calculations.

To compute V_{obs} on the U scale we need to derive DM^{ECM}, DM^{CM}, and $V_{\text{aug}} \equiv \mathsf{Var}(U^\star|Y^\star)$, where $Y^\star = \mathsf{E}(Y|\theta^\star, Y_{\text{obs}})$. Implementing `Algorithm 2` on the mapping induced by M_U will yield DM^{ECM}. Since this is not a PECM algorithm, we cannot use (4.2.3) to derive DM^{CM}. Instead, SCM in `Algorithm 3` is used. Replace $Y^{(t)} = \{Y_{ijk}^{(t)}, i, j, k = 1, 2\}$ with $Y^\star$ in each CM-step and use the CM algorithm iteration, $\text{CM}_1 \rightarrow \text{CM}_2 \rightarrow \text{CM}_3$ in `CMSTEPS`; `Algorithm 3` differentiates this mapping and produces DM^{CM}. Finally, we compute V_{aug} via a log-linear models package. Since all standard programs use the sufficient statistics as their input, and I_{aug} is linear in the augmented-data sufficient statistics, fitting (4.4.1) using $Y^\star$ as the data will yield V_{aug}. For example, in *'S'* (AT&T Bell Laboratories) V_{aug} can be computed with

```
> model<-glm(formula= Y* ~ P + C + S + PC + PS + CS,
  family = poisson(link = log))

>  summary(model)$cov.unscaled
```

where `P`, `C`, `S`, etc. are vectors of $+1$ and -1 determined by (4.4.1) and are the columns of the design matrix X. The parameter u_0 in (4.4.1) is just a scale parameter that insures that $\sum \theta_{ijk} = 1$ and it should be ignored in the calculation of V_{obs} as there are only six free parameters. Replace DM^{ECM}, DM^{CM}, and V_{aug} with the 6×6 submatrices corresponding to the other six parameters before computing V_{obs} using (4.2.5).

The results are presented in Table 4.5. The calculated matrix V_{obs} was symmetric to nine places beyond the decimal, which is more accurate than expected

	u_P	u_S	u_C	u_{PS}	u_{CS}	u_{PC}
$U^\star$ †	0.406944871	−1.565681190	0.181533221	−0.044421642	−0.424777146	−0.661665499
$\mathrm{sd}(U^\star)$	0.117611832	0.092744286	0.135159680	0.117485008	0.132665503	0.058468375
V_{obs}	0.013832543	−0.001820669	0.010260487	0.012248728	0.009021917	−0.001188741
	−0.001820668	0.008601503	0.003117087	−0.002317203	0.002471191	−0.000336477
	0.010260487	0.003117086	0.018268139	0.008670460	0.016175245	−0.001602312
	0.012248728	−0.002317204	0.008670459	0.013802727	0.009881752	0.001145066
	0.009021918	0.002471190	0.016175245	0.009881753	0.017600136	0.000532728
	−0.001188741	−0.000336477	−0.001602312	0.001145066	0.000532728	0.003418551

† $u_0^\star = -3.329440804$

Table 4.5. The MLE $U^\star$ and V_{obs} for model (4.4.1) with data given in Table 4.4

.

		Survival (S)	
Clinic (C)	Prenatal Care (P)	Died	Survived
A	Less	[0.0016, 0.0115]	[0.2244, 0.2876]
	More	[0.0040, 0.0156]	[0.3566, 0.4200]
B	Less	[0.0180, 0.0391]	[0.2531, 0.3181]
	More	[0.0013, 0.0090]	[0.0205, 0.0457]

Table 4.6. 95% marginal confidence intervals for $\theta^\star$ in model (4.4.1).

since the algorithm was run with $\xi^{ECM} = 10^{-16}$, $\xi^{SECM} = 10^{-8}$, and $\xi^{SCM} = 10^{-7}$. The ECM algorithm required 70 iterations to converge.

The information in Table 4.5 can be used to construct confidence intervals. For example, we can derive the Jacobian J of the transformation from $\text{logit}(\theta)$ to U in order to calculate the observed Fisher information matrix for $\text{logit}(\theta^\star)$: $I_o(\text{logit}(\theta^\star)|Y_{\text{obs}}) = J^\top I_o(U^\star|Y_{\text{obs}})J$. Assuming approximate normality on the $\text{logit}(\theta)$ scale, we can construct confidence intervals for each of $\text{logit}(\theta_{ijk})$, and then transform to the θ scale. Table 4.6 is an illustration.

The calculations presented in the context of this example are in fact quite general. Bishop, Feinberg, and Holland (1975) describes how either IPF or closed form solutions can be used to fit any hierarchical log-linear model to contingency tables with augmented data. This means that the CM-steps can easily be identified for any such model. Since the E-step in (4.4.2) can easily be generalized to any table with incomplete data, the SECM algorithm for fitting a log-linear model to any table can easily be formulated. These calculations are described more fully by van Dyk (1993).

Y_1	8	6	11	22	14	17	18	24	19
Y_2	59	58	56	53	50	45	43	42	39
Y_1	23	26	40	4	4	5	6	8	10
Y_2	38	30	27	-	-	-	-	-	-

Table 4.7. The data used in the monotone missing-data example (Section 4.4.5.

4.4.5. A bivariate normal example

The data given in Table 4.7 are assumed to follow a bivariate normal distribution with parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, \rho)$. As in Meng and Rubin (1991a), we will use the parameterization $\theta = (\mu_1, \mu_2, \log\sigma_1, \log\sigma_2, Z_\rho)$. Notice that because of the monotone pattern of missing values, there is no missing information for $\vartheta_1 = (\mu_1, \sigma_1)$, and hence we can use either the standard ECM algorithm, or first estimate ϑ_1 to get $\vartheta_1^\star$ using standard augmented-data estimates and then use ECM* of Section 4.2.4 to find $\vartheta_2^\star$. To implement the standard ECM algorithm, we will use the the two CM-steps

$$\begin{aligned} \text{CM}_1: \quad & \vartheta_1^{(t+\frac{1}{2})} = CM(\vartheta_1 \,|\, \vartheta_2^{(t)}) \\ \text{CM}_2: \quad & \vartheta_2^{(t+\frac{2}{2})} = CM(\vartheta_2 \,|\, \vartheta_1^{(t+\frac{1}{2})}), \end{aligned} \tag{4.4.4}$$

where $\vartheta_2 = (\mu_1, \log\sigma_1, Z_\rho)$, and the function $CM(\xi|\phi)$ takes on the value of ξ that maximizes $Q(\theta|\theta^{(t)})$ conditional on ϕ. To implement ECM*, we will fix $\vartheta_1 = \vartheta_1^\star$ and use the algorithms in Section 4.3. We need only note that the input for `ESTEP` is $\theta^{(t)} = (\vartheta_1^\star, \vartheta_2^{(t)})$ and `CMSTEPS` consists of the CM$_2$-step given in (4.4.4). The increase in variance due to the missing data, ΔV, can be calculated with (4.2.8)

where $\Delta V(\vartheta_2^\star|\vartheta_1^\star)$ is given by (4.2.7). DM^{ECM^*} and $\Delta V(\vartheta_2^\star|\vartheta_1^\star)$ are given in Table 4.8 for several values of ξ^{ECM} and ξ^{SECM}.

It is interesting to compare ECM* with EM. Since there is no missing information for ϑ_1, in this example EM produces $\vartheta_1^\star$ in one step. Consequently, the E-steps from EM after one iteration and form ECM* both condition on $\vartheta_1 = \vartheta_1^\star$ and will produce the same sufficient statistics, $S^{(t)}(Y_{\text{obs}})$ for the same value of $\vartheta_2^{(t)}$. It is a simple fact of calculus in this example that $CM(\vartheta_2|\vartheta_1 = \vartheta_1^*) = CM(\vartheta_2)$. The maximization step from EM and ECM* will produce the same value of $\vartheta_2^{(t+1)}$ for the same value of $S^{(t)}(Y_{\text{obs}})$ after one iteration. Therefore, the mapping induced on ϑ_2 by running ECM* will be the same as the mapping induced on ϑ_2 by running EM. Comparing DM^{ECM^*} given in Table 4.8 with $DM^\star$ given in Meng and Rubin (1991a) verifies this. There is a slight difference between the number of iterations required because we used the convergence criterion (i) of Section 4.3.4 whereas they used (iii).

Since EM and ECM* induce the same mappings in this situation, SEM and SECM* are identical. SECM has a distinct advantage over SEM, however. When there is no missing information for ϑ_1, the standard SEM algorithm fails because some of the denominators of (4.3.3) are zero and therefore a special version similar to SECM* must be implemented (Meng and Rubin, 1991a). Since the CM algorithm does not converge in one iteration even when there is no missing information, the denominators are never zero, and thus the situation is less critical for SECM and we may proceed with the standard algorithm. Table 4.9 records the results of applying the standard SECM algorithm given in (4.4.4). As is evident from Ta-

Stopping Criteria ECM;SECM	ECM Iterations for $\vartheta_2^\star$	SECM Iterations for DM^{ECM^*}			DM^{ECM^*}			$\Delta V(\vartheta_2^\star \vert \vartheta_1^\star)$		
$10^{-4};\ 10^{-2}$	25	2	3	2	0.333	0.042	−0.052	1.151285	0.138119	−0.232521
		7	6	6	1.407	0.289	0.010	0.170981	0.025525	−0.024140
		11	5	3	−0.648	0.017	0.470	−0.110795	−0.008875	0.039972
$10^{-8};\ 10^{-4}$	44	2	11	14	0.333	0.040	−0.028	1.085282	0.166673	−0.093734
		19	17	16	1.444	0.299	0.019	0.167022	0.028514	−0.009820
		20	12	17	−0.642	0.015	0.325	−0.093318	−0.009753	0.019379
$10^{-12};\ 10^{-6}$	63	2	20	23	0.333	0.050	−0.028	1.085838	0.167083	−0.093349
		29	26	26	1.444	0.299	0.019	0.167087	0.028555	−0.009778
		29	20	26	−0.642	0.015	0.325	−0.093344	−0.009777	0.019352

Table 4.8. Number of ECM * and SECM * iterations and convergent values of DM^{ECM^*} and $\Delta V(\vartheta_2^\star|\vartheta_1^\star)$ under three different convergence threshold in the monotone missing-data example (Section 4.4.5).

Stopping Criteria ECM; SECM	ECM Iterations for $\theta^\star$	SECM Iterations for DM^{ECM}					ΔV				
10^{-4}; 10^{-2}	47	3	2	3	2	2	-0.216878	-0.026844	0.509994	0.034717	0.183486
		4	3	3	3	3	0.008955	0.008688	-0.006716	0.004098	-0.014324
		6	2	5	2	2	0.343573	0.060970	0.331937	0.125992	-0.410751
		5	6	7	5	6	0.023331	0.012508	0.135287	0.032239	-0.035117
		10	7	11	7	5	-0.009776	-0.011889	-0.088636	-0.015762	0.043422
10^{-8}; 10^{-4}	88	8	6	8	7	8	0.022126	-0.001376	-0.116788	-0.003779	0.031776
		23	19	8	18	20	0.000889	-0.001061	-0.003298	-0.001884	0.000524
		26	9	23	13	20	-0.019489	-0.002430	1.196050	0.165772	-0.125586
		7	21	23	7	22	0.001040	-0.001843	0.162477	0.025473	-0.009010
		23	19	23	9	22	-0.000522	0.000661	-0.091476	-0.008647	0.019110
10^{-12}; 10^{-6}	130	44	31	41	34	41	-0.000001	0.000076	-0.000046	0.000073	0.000044
		35	32	29	31	32	0.000001	0.000000	-0.000003	-0.000001	0.000000
		47	31	44	34	41	0.000027	-0.000094	1.085844	0.166995	-0.093399
		28	33	35	31	34	0.000003	-0.000001	0.167081	0.028554	-0.009779
		35	31	35	28	34	-0.000002	0.000000	-0.093342	-0.009777	0.019352

Table 4.9. Number of ECM and SECM iterations and convergent values of ΔV under three different convergence thresholds in the monotone missing-data example (Section 4.4.5).

ble 4.8, however, the standard SECM algorithm requires approximating the zero elements in the rate matrix and is therefore less efficient and less stable. In particular, SECM is slower to converge, requires smaller values of ξ^{ECM} and ξ^{SECM} to obtain satisfactory symmetry, and requires the calculation of the DM^{ECM} matrix, which can be of a much larger dimension than the DM^{ECM^*} matrix. Nevertheless, it is noteworthy that standard SECM does produce V_{obs}, whereas SEM must be modified in this situation.

4.5. Diagnostics and Variations

4.5.1. Checking the symmetry of V_{obs}

One of the most valuable properties of SECM is that, like SEM, it has a built-in diagnostic. Section 4.3 describes all the steps required by SECM. It is the last step, the computation of ΔV, that helps us know if mistakes have been made in any of these steps. The variance-covariance matrix, $V_{\text{obs}} = V_{\text{aug}} + \Delta V$, and thus ΔV, must be symmetric, but if any of $\theta^\star$, DM^{ECM}, DM^{CM}, or I_{aug} are not calculated correctly, it is practically certain that the resulting ΔV and hence V_{obs} will be asymmetric. The example in Section 4.4.1 documents that this diagnostic not only checks the computation of V_{obs} but also detects errors in implementing the E and CM-steps. Convergence of the $\theta^{(t)}$ sequence does not insure that the convergent value is the MLE, $\theta^\star$. Many erroneous algorithms converge. In fact, we had an instance in which our algorithm increased the likelihood

at each step and converged, but the resulting V_{obs} was asymmetric. In this case, careful checking led to the discovery of some subtle errors in implementation. There is no other diagnostic known to us that can automatically detect these errors, and one perhaps would never find them without the detection power of this tool. If V_{obs} is symmetric, however, we are virtually assured that both $\theta^\star$ and V_{obs} are correct because it seems practically impossible to make separate errors in SECM that cancel appropriately.

Even when SECM is implemented correctly, the convergence threshold ξ^{ECM} needs to be quite small in order to obtain a V_{obs} matrix with satisfactory symmetry; this implies an increase in the number of iterations required, especially when θ is of high dimension. The more precisely we calculate $\theta^\star$, the more accurately we are able to compute V_{obs}, because we are able to compute DM^{ECM} and DM^{CM} more accurately, as demonstrated by the example in Section 4.4.5. In general ξ^{SECM} and ξ^{SCM} are chosen to be about equal to the square root of ξ^{ECM}. They should be chosen as small as possible, however, so that (4.3.4) is satisfied for some t_{ij} for each i and j. In order to increase the accuracy of DM^{ECM} and DM^{CM}, ξ^{SECM} and ξ^{SCM} may even be different for different components. When deciding on convergence thresholds, a good rule of thumb is that V_{obs} will be symmetric to about half as many digits as $\theta^\star$ is precise; for example, roughly, when ξ^{ECM} is 10^{-8}, we can expect 3 or 4 digits of accuracy in V_{obs}. The accuracy of V_{obs} can always be judged by its symmetry, however, and gross asymmetry always indicates either errors in implementation or numerical imprecision; also see Section 5 of Meng and Rubin (1991a).

4.5.2. Computing V_{obs} when implementing AECM

The AECM algorithm described in Chapter 3 is a generalization of ECM in which the data-augmentation and model-reduction schemes are allowed to vary from iteration to iteration. Although this can lead to much faster algorithms, the mathematical formulation of the rate matrix is much more complicated than that of ECM (see Section 3.3). Consequently, direct implementation of the supplemented AECM algorithm would be quite involved. Sometimes, however, there is an easy solution. In `Algorithms 2 and 3`, we evaluate the ratio r_{ij} using the ECM iterates in order to calculate DM^{ECM} and DM^{CM}. But there is nothing that requires the use of the ECM iterates in these algorithms, just the ECM code, and we can actually use any sequence $\theta^{(t)}$ converging to $\theta^{\star}$. For the AECM algorithm there is generally a corresponding ECM mapping (i.e., from the first cycle of each AECM iteration). Such an ECM mapping may not be useful for computing $\theta^{\star}$ because it may not be space filling, but it can be used for the SECM algorithm which does not require the space-filling condition once $\theta^{\star}$ is obtained. In the case of MCECM, we can also obtain the ECM code by simply dropping all but the first E-step for each space-filling sequence of CM-steps. With the ECM code in hand, we can use the AECM iterates as input for the algorithms and compute V_{obs} just as described in Section 4.3. That is, AECM can be used when we calculate $\theta^{\star}$ in `Algorithm 1`, but the ECM algorithm should be used in `Algorithms 2` and `3` in which `ESTEP` will consist of one E-step, and `CMSTEPS` will consists of a space-filling set of CM-steps. If the AECM iterations were not saved, we can simply run the ECM algorithm when

implementing the supplemented algorithm to calculate V_{obs}. Generally, running ECM in this round will not slow the convergence, since we only need to start at initial values that are close to $\theta^{\star}$.

Chapter 5

Efficient Model Reduction: Permuting CM Steps

5.1. Introduction

The AECM algorithm developed in Chapter 3 can increase the efficiency of the EM algorithm via a more flexible data-augmentation scheme and simplifies its implementation by reducing the augmented-data model. Chapter 2 developed a method of selecting an efficient data-augmentation scheme. In the current chapter, we will examine the possibility of building more efficient algorithms by looking at a key aspect of model reduction in the context of the ECM algorithm. Specifically, with ECM, model reduction extends the EM algorithm by replacing the M-step with several CM-steps. Because each of the CM-steps maximizes $Q(\theta|\theta^{(t)})$ over a different subspace of Θ, the "space-filling" condition is required to guarantee that the whole parameter space Θ will be searched after we perform a set of CM-steps, but there is no restriction on the order in which we choose to perform them. Moreover, the order of the CM-steps can significantly affect the number of iterations required for

convergence. Taking an extreme example, when the ECM algorithm was used to fit a log-linear model to a certain sparse contingency table with partially classified counts, the ECM algorithm with one order of CM-steps required 27 times as many iterations (i.e., CPU time) as another order. It is therefore of practical interest to conduct an investigation of the impact of ordering on the computation time required by ECM, especially since the same computations are required to implement one iteration of ECM regardless of ordering.

The purpose of this chapter is twofold. First, using the common contingency table problem, we illustrate the impact of changing orderings on the actual number of steps required for convergence. Second, we bring to light several remarkable and sometimes troubling phenomena that arose in our investigation. We will see how the standard theory for studying such algorithms and the seemingly related theory of stochastic iterative algorithms lead to conclusions that are quite different from what we see in practice. At a very practical level, we will show that the missing-data structure seems to be ancillary to the relative efficiency of CM-step orderings and that in some situations one of the standard convergence criteria can be very misleading. These findings reinforce a lesson that is often forgotten — empirical study is an irreplaceable criterion for realistic evaluation. It is somewhat remarkable that this lesson repeats itself again and again in one study, where theory, intuition and common wisdom all fail the empirical evaluation.

In order to evaluate the effect of permuting CM-steps, the global rate of convergence for ECM is introduced and developed in Section 5.2. The difference between theory and empirical findings is reported in Section 5.3, which discusses

reversing the order of CM-steps, and an investigation of general permutations is presented in Section 5.4. Section 5.5 explores the sources of variation in iterations required for convergence. In light of the theory of stochastic relaxation, Section 5.6 studies the question of whether it is a good strategy in practice to cycle through the different permutations of CM-steps or to randomly select an order at each iteration. Section 5.7 provides concluding remarks including several interesting practical issues in selecting a convergence criterion.

5.2. The Rate of Convergence of ECM

5.2.1. The matrix rate

In Section 1.6 we presented the matrix and global rates of convergence for EM. In the development of the SECM algorithm, the matrix rate was extended to ECM in Section 4.2.2. In this section, we will extend the global rate to ECM. Like EM, the ECM algorithm implicitly defines a mapping $M^{ECM} : \theta^{(t)} \rightarrow \theta^{(t+1)} = M^{ECM}(\theta^{(t)})$ from the parameter space Θ to itself. Let $\theta^{\star}$ be the limit of $\{\theta^{(t)} : t \geq 0\}$. Suppose that $M^{ECM}(\theta)$ is differentiable in a neighborhood of $\theta^{\star}$. Then a Taylor's series approximation yields

$$(\theta^{(t+1)} - \theta^{\star}) \approx (\theta^{(t)} - \theta^{\star}) DM^{ECM}(\theta^{\star})$$

where

$$DM^{ECM}(\theta) = \left(\frac{\partial M_j^{ECM}(\theta)}{\partial \theta_i} \right),$$

and $DM^{ECM}(\theta^\star)$ is nonzero when there is missing information (Dempster, Laird, and Rubin, 1977; Meng and Rubin, 1991a, 1994a) or when there is more than one CM-step. Thus, the ECM mapping is linear if we ignore the higher order terms in the Taylor's series expansion. This approximation becomes exact at convergence of ECM, and thus $DM^{ECM}(\theta^\star)$ is called the (matrix) rate of convergence of ECM (e.g., Meng, 1994). Since $DM^{ECM}(\theta)$ will always be evaluated at $\theta = \theta^\star$, we will suppress the dependency on θ when referring to the matrix rate of convergence.

5.2.2. The global rate

As with EM, in order to assess convergence, consider the ratio

$$r_t = \frac{||\theta^{(t)} - \theta^\star||}{||\theta^{(t-1)} - \theta^\star||}, \quad t = 1, 2, \ldots \tag{5.2.1}$$

where $||\cdot||$ is the Euclidean norm. Algorithms which have smaller values of r_t tend to converge more quickly. It is trivial to show that

$$\lim_{t\to\infty} \left| \sqrt[t]{||\theta^{(t)} - \theta^\star||} - \left[\prod_{i=1}^{t} r_i\right]^{\frac{1}{t}} \right| = 0. \tag{5.2.2}$$

The lim sup of the term $\rho_t = \sqrt[t]{||\theta^{(t)} - \theta^\star||}$ is known as the root convergence factor (denoted ρ) and is equal to the spectral radius of DM^{ECM} (i.e., the largest norm of the eigenvalues; Ortega and Rheinboldt, 1970, 10.1.4). Equation (5.2.2) says that the geometric mean of the r_t tends to the spectral radius of DM^{ECM}. Thus, the spectral radius of DM^{ECM} controls the rate of convergence of the ECM algorithm, and, in theory, algorithms with smaller spectral radii are preferred. For computational purposes, we will use the empirical root convergence factor $\hat{\rho}_t = [\prod_{i=2}^{t} \hat{r}_i]^{1/(t-1)}$, where $\hat{r}_i = ||\theta^{(i)} - \theta^{(i-1)}||/||\theta^{(i-1)} - \theta^{(i-2)}||$, $i \geq 2$.

It is worth noting that if DM^{ECM} has a spectral decomposition under similarity transformations (as is always the case with the EM mapping; see Meng and Rubin, 1994a), $r = \lim_{t\to\infty} r_t$ exists and is equal to the spectral radius of DM^{ECM}. Thus, in the EM literature (as in Section 1.6) the convergent value of (5.2.1) is used to assess the rate of convergence of the algorithm. This will not suffice in the ECM case as $\lim_{t\to\infty} r_t$ may not exist and $\limsup_{t\to\infty} r_t$ can be greater than the spectral radius. This is illustrated in Figures 5.1 and 5.2 with a pure linear iteration of a particular DM^{ECM} matrix. In the figures, θ is three dimensional and the componentwise rate of convergence is defined as $r_{t,i} = (\theta_i^{(t)} - \theta_i^\star)/(\theta_i^{(t-1)} - \theta_i^\star)$, $i = 1, 2, 3$ (i.e., (5.2.1) applied to each component).

In practice, the quantity of real interest is the actual number of iterations an algorithm requires for convergence. If $\hat{\rho}_t$ can be well approximated by ρ, the spectral radius of DM^{ECM}, then the number of iterations required for convergence, N, is related to ρ via

$$N \propto \frac{-1}{\log(\rho)}, \qquad \textbf{(5.2.3)}$$

where the constant of proportionality depends only on the starting value and the convergence criterion.

5.2.3. Factors effecting the global rate

In order to study ρ under different orderings of CM-steps, we need to investigate how DM^{ECM} varies with these orderings. As discussed in Chapter 1, Meng (1994) showed that

$$[I - DM^{ECM}] = [I - DM^{EM}][I - DM^{CM}], \qquad \textbf{(5.2.4)}$$

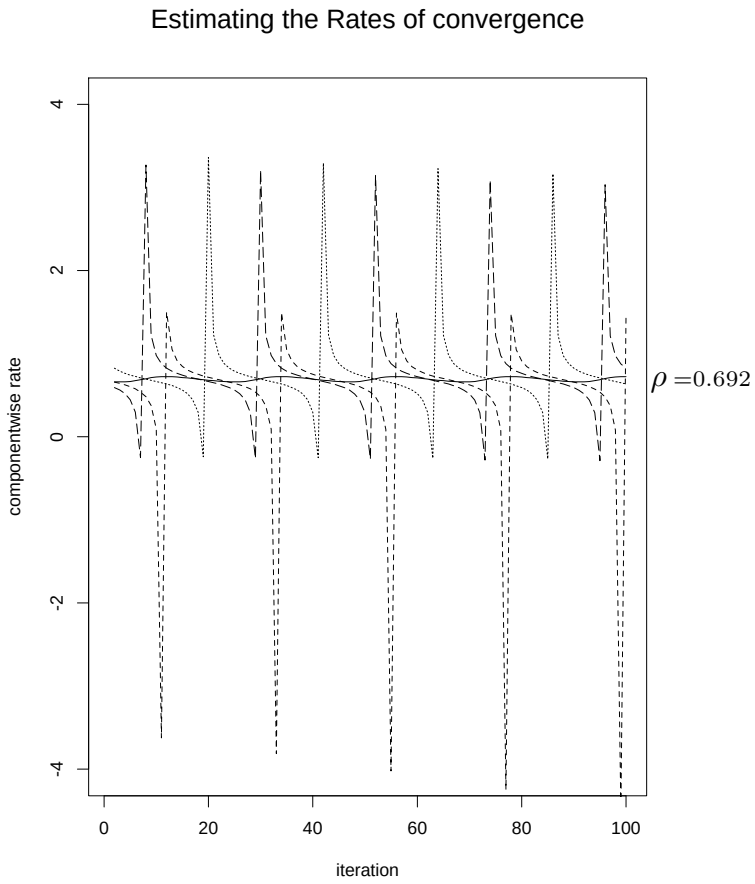

Figure 5.1. Rates of convergence with a complex dominant eigenvalue. The plot shows the overall rate of convergence (5.2.1) as a function of the iteration (solid line). The dashed lines are (5.2.1) applied to components of θ.

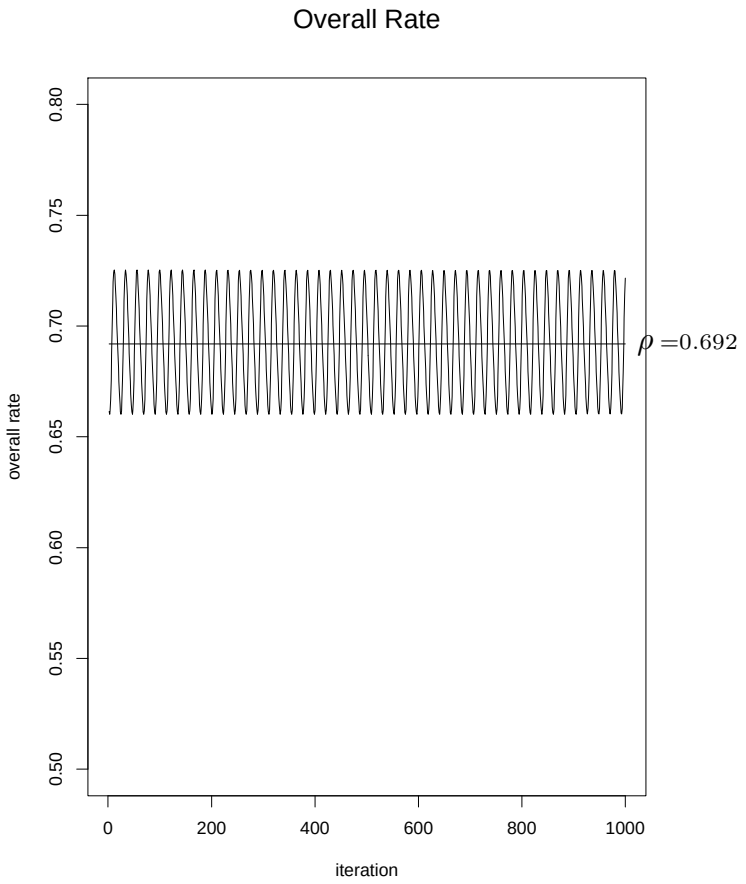

Figure 5.2. The empirical root convergence factor for large t. The plot shows that the oscillation of (5.2.1) around ρ (the straight line) does not dampen. Note that $\limsup r_t > \rho = 0.692$.

where I is the identity matrix, $DM^{EM} = I - I_{\text{obs}} I_{\text{aug}}^{-1}$, and

$$DM^{CM} = P_1 \cdots P_S, \quad \text{with} \quad P_s = \nabla_s [\nabla_s^\top I_{\text{aug}}^{-1} \nabla_s]^{-1} \nabla_s^\top I_{\text{aug}}^{-1} \quad \text{and} \quad \nabla_s = \nabla g_s(\theta^\star). \tag{5.2.5}$$

In the above expressions, DM^{EM} and DM^{CM} denote the matrix rate of convergence of the EM and CM algorithms respectively. From (5.2.4) – (5.2.5), we see that DM^{ECM} depends on the data augmentation through DM^{EM} and on the model reduction scheme (in particular, the order of CM-steps) through DM^{CM} where the product of P_s 's is calculated in the order that the CM-steps are performed.

There is another way of expressing DM^{CM} that turns out to be theoretically convenient:

$$DM^{CM} = I_{\text{aug}}^{\frac{1}{2}} \{A_1 \cdots A_S\} I_{\text{aug}}^{-\frac{1}{2}}, \tag{5.2.6}$$

where

$$A_s = \xi_s [\xi_s^\top \xi_s]^{-1} \xi_s^\top \quad \text{with} \quad \xi_s = I_{\text{aug}}^{-\frac{1}{2}} \nabla_s.$$

In other words, DM^{CM} can be written as the product of symmetric (projection) matrices.

5.3. Reversing the Order of CM-Steps

5.3.1. Theoretical results

Consider the two ECM algorithms, ECM $_{(1,\ldots,S)}$ with steps ordered

$$E \to CM_1 \to \cdots \to CM_S,$$

and ECM $_{(S,\ldots,1)}$ with steps

$$E \to CM_S \to \cdots \to CM_1.$$

In general, we will use the notation ECM $_\alpha$ to denote the ECM algorithm with CM-steps ordered as in the ordered set α (e.g., $\alpha = (1,2,4,3)$). This section is devoted to comparing ECM $_{(1,\ldots,S)}$ and ECM $_{(S,\ldots,1)}$, the effect of reversals. The following result asserts that the eigenvalues of $DM^{ECM_{(1,\ldots,S)}}$ and $DM^{ECM_{(S,\ldots,1)}}$ are identical, a result that is not immediately intuitive.

Theorem 5.1: In the ECM algorithm, reversing the CM-steps has no effect on the eigenvalues of DM^{ECM} .

Proof: From (5.2.4) – (5.2.6),

$$\begin{aligned}[I - DM^{ECM_{(1,\ldots,S)}}] =& [I - DM^{EM}][I - I_{\text{aug}}^{\frac{1}{2}} (A_1 \cdots A_S) I_{\text{aug}}^{-\frac{1}{2}}] \\ =& I_{\text{obs}} I_{\text{aug}}^{-1} [I - I_{\text{aug}}^{\frac{1}{2}} (A_1 \cdots A_S) I_{\text{aug}}^{-\frac{1}{2}}] \\ =& I_{\text{obs}} [I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}} (A_1 \cdots A_S) I_{\text{aug}}^{-\frac{1}{2}}]. \qquad (\mathbf{5.3.1})\end{aligned}$$

Since transposition does not change eigenvalues, (5.3.1) has the same eigenvalues as

$$\left\{ I_{\text{obs}} [I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}} (A_1 \cdots A_S) I_{\text{aug}}^{-\frac{1}{2}}] \right\}^\top = \left\{ I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}} (A_S \cdots A_1) I_{\text{aug}}^{-\frac{1}{2}} \right\} I_{\text{obs}}. \qquad (\mathbf{5.3.2})$$

Finally, using the fact that for square matrices G and H , GH and HG have the same eigenvalues, we can conclude from (5.3.1) – (5.3.2) that $I - DM^{ECM_{(1,\ldots,S)}}$ has the same eigenvalues as

$$I_{\text{obs}}[I_{\text{aug}}^{-1} - I_{\text{aug}}^{-\frac{1}{2}}\,(A_S \cdots A_1)\,I_{\text{aug}}^{-\frac{1}{2}}] = [I - DM^{EM}][I - DM^{CM_{(S,\ldots,1)}}]$$
$$= I - DM^{ECM_{(S,\ldots,1)}}.$$

■

From the discussion in Section 5.2, Theorem 5.1 tells us that reversing CM-steps within the ECM algorithm does not effect the root convergence factor of the algorithm. We must recall, however, that although ECM tends to be linear near $\theta^\star$, little is known about its behavior away from $\theta^\star$. Moreover, the empirical root convergence factor, $\hat{\rho}_t$, is only asymptotically (with respect to t) equal to the spectral radius of DM^{ECM}. So even if the algorithm is linear, it may take many iterations before $\hat{\rho}_t$ converges to the spectral radius.

5.3.2. Empirical results

To judge the applicability of Theorem 5.1, a simulation was performed. The ECM algorithm was used to fit a log-linear model to data from a partially classified $2 \times 2 \times 2$ contingency table (see Sections 3.5.2 and 4.4.4). This ECM algorithm has three CM-steps, which correspond to the three steps of iterative proportional fitting (IPF, Bishop, Feinberg, and Holland, 1975); the details are given in Section 4.4.4. For each simulation 2000 data sets of size n were generated, of which n_c were completely classified and n_i were classified only according to margin i ($i = 1, 2, 3$). Three simulations were run and the sample sizes for each appear in Table 5.1. Each data set was generated from the no three-way interaction model

Sim.	n	n_c	n_1	n_2	n_3
1	100	25	25	25	25
2	1000	100	300	300	300
3	100	40	20	20	20

Table 5.1. Sample sizes for simulations 1-3. This table records the total number of observations n, the number of observations that are completely classified n_c, and the number n_i which were classified only according to margin i.

	Cyclic Permutations		
Reversals of	ECM $_{(123)}$	ECM $_{(231)}$	ECM $_{(312)}$
CM Steps	ECM $_{(321)}$	ECM $_{(132)}$	ECM $_{(213)}$

Table 5.2. The ECM algorithms run in the simulations.

$$\log(\theta_{ijk}) = u_0 + (-1)^{i-1}u_1 + (-1)^{j-1}u_2 + (-1)^{k-1}u_3 + (-1)^{i+j}u_{12} + (-1)^{j+k}u_{23} + (-1)^{i+k}u_{13}, \qquad (\mathbf{5.3.3})$$

where θ_{ijk} is the cell probability for cell (i,j,k). As an attempt to mimic the variability in parameters governing real data sets, the log-linear parameters were randomly selected for each data set from $u_l \sim N(0,1)$, and $u_{lm} \sim N(0,0.25)$, independently for $l,m = 1,2,3$, and u_0 was then chosen so that the cell probabilities sum to one.

For each data set the model in (5.3.3) was fit using a starting value of $\theta_{ijk}^{(0)} = \frac{1}{8}$, which is frequently used in practice, with each of the six ECM algorithms resulting from the six possible orderings of the CM-steps. Suppose N_{ij} $(i = 1,2;\ j = 1,2,3)$ are the number of steps required for convergence of the six algo-

rithms where N_{1j} correspond to three algorithms with different asymptotic rates of convergence (c.f., columns of Table 5.2), and N_{i1} correspond to the algorithms with the CM-steps reversed (c.f., rows of Table 5.2). Consider the quantity,

$$R^2 = \frac{\frac{1}{2}\sum_i (N_{1j} - N_{2j})^2}{\sum_{ij}(N_{ij} - \bar{N}_{..})^2}, \qquad (\mathbf{5.3.4})$$

which is the proportion of the total variation among the N_{ij} that can be accounted for by reversing the CM-steps. If Theorem 5.1 also implies that reversing the order of CM-steps does not effect the actual number of steps required for convergence, R^2 should be near zero. Figure 5.3 shows that simulated values of R^2 from the three simulations are actually skewed towards one (computed when the denominator of (5.3.4) was greater than zero). Thus, step reversals seem to be more important than Theorem 5.1 implies.

To explore this further, Figure 5.4 shows estimated densities for the possible percent increase in efficiency due to (a) step reversals (dotted line: $1 - \min(N_{1j}, N_{2j})/\max(N_{1j}, N_{2j})$) and (b) general permutations (solid line: $1 - \min_{i,j}(N_{ij})/\max_{i,j}(N_{ij})$). Notice that although the effect of general step permutations is slightly greater than that of simple step reversals, the step reversals account for the majority of the variation. This implies that when we consider the effect of permutation of CM-steps on CPU time, we must consider all $S!$ possible orderings (as will be discussed in the following section). More than this, however, we must be somewhat skeptical of analysis of convergence properties of EM-type algorithms based solely on the spectral radius of the mapping matrix. Such analysis is much better defined mathematically and more widespread in the literature than attempts to look at the actual CPU time required for convergence, but may not answer

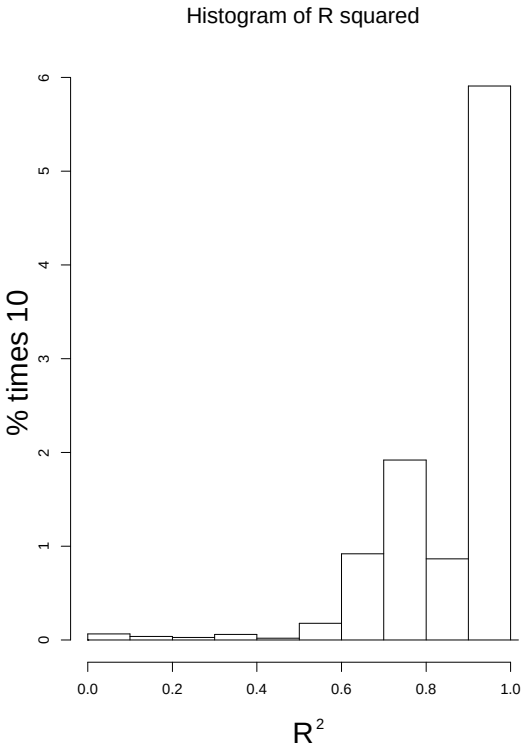

Figure 5.3. A histogram of the 6000 values of R^2 observed in the three simulations described in Table 5.1. Notice that R^2 tends to be large which sheds doubt on the applicability of Theorem 1 to the number of steps required for convergence.

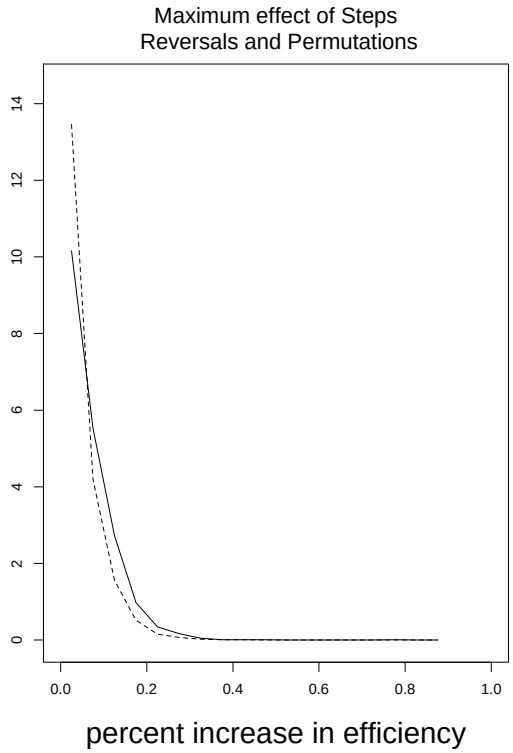

Figure 5.4. The figure shows the density of percent decrease in iterations required due to (a) step reversals: $[\,1 - \min(N_{1j}, N_{2j})/\max(N_{1j}, N_{2j})\,]$ (dotted line); and (b) general permutation: $[\,1 - \min(N_{ij})/\max(N_{ij})\,]$ (solid line).

questions that users actually face. (An even more relevant comparison in practice should take into account the users' effort for implementing different algorithms, an issue that fortunately need not be considered here because implementing a different ordering does not increase users' effort). For completeness, in the next section, we will present several results which bound the effect of permuting CM-steps on the spectral radius, but the rest of the chapter will focus on on the actual CPU time, or equivalently when making comparisons, the actual number of iterations.

5.4. General Permutations of CM-steps

5.4.1. Theoretical results

For an ECM algorithm with S CM-steps, there are $S!$ possible orders. In this section we will explore how much difference there is between these $S!$ algorithms with regard to CPU time. Again, we will begin our discussion with a look at the root convergence factor and compare this with results from the simulation presented in Section 5.3.

Theorem 5.2 will help us to put a bound on the effect of permuting CM-steps on the spectral radius of DM^{ECM} in the three CM-step problem.

Theorem 5.2: For the general three CM-step ECM algorithm, the determinant of $I - DM^{ECM}$ is invariant under permutations of CM-steps.

Proof: First we observe that in the three step CM algorithm, the eigenvalues of DM^{CM} do not depend on the order of CM-steps. To see this, note (i) CM is a special case of ECM with $I_{\text{obs}} = I_{\text{aug}}$, so the observation holds under CM-step reversals (Theorem 5.1); (ii) it is clear from (5.2.6) that the observation holds under cyclic permutations (for ECM the eigenvalues will generally change with cyclic permutations of CM-steps because of the E-step). Together (i) and (ii) give all possible permutations in the three step algorithm. From this observation it is clear that $|I - DM^{CM}|$ is constant over permutations of CM-steps. Finally, note

that from (5.2.4), we have

$$|I - DM^{ECM}| = |I_{\text{obs}}|\,|I_{\text{aug}}^{-1}|\,|I - DM^{CM}| \qquad (\mathbf{5.4.1})$$

The result is now clear since none of the quantities on the right are effected by permutation of CM-steps.

■

For the general ECM algorithm, Theorem 5.2 holds only when $S = 3$. For the important special case of PECM, however, the result holds for any S.

Theorem 5.3: For the PECM algorithm (see for example Section 1.3),

$$|I - DM^{CM}| = \frac{|I_{\text{aug}}|}{\prod_{i=1}^{S}|\Upsilon_i|} \quad \text{and} \quad |I - DM^{ECM}| = \frac{|I_{\text{obs}}|}{\prod_{i=1}^{S}|\Upsilon_i|},$$

where Υ_i are the submatrices of the block diagonal matrix Υ (4.2.3). Both are invariant under permutations of CM-steps.

Proof: From (4.2.3), we can write

$$[I - DM^{CM}] = [\Upsilon + \Gamma^{\top} + \Gamma][\Upsilon + \Gamma^{\top}]^{-1} = I_{\text{aug}}[\Upsilon + \Gamma^{\top}]^{-1}.$$

Since $[\Upsilon + \Gamma^{\top}]$ is a block triangular matrix, with diagonal submatrices Υ_i, its determinant is $\prod_{i=1}^{S}|\Upsilon_i|$, which gives the result for $|I - DM^{CM}|$. Again, (5.4.1) easily extends this to $|I - DM^{ECM}|$.

■

The following corollary relates Theorems 5.2 and 5.3 to the spectral radius of DM^{ECM}.

Corollary: Let $\rho_\alpha \leq \rho_\beta$ be the spectral radii of DM^{ECM_α} and DM^{ECM_β} respectively, where ECM $_\alpha$ and ECM $_\beta$ are either both PECM or both three step ECM

algorithms, and in either case, both algorithms are identical except for the order of the CM-steps. Suppose

$$|I - DM^{ECM_\beta}| \leq 1 - \rho_\beta, \tag{5.4.2}$$

then

$$\rho_\beta \geq \rho_\alpha \geq 1 - \sqrt[d]{1 - \rho_\beta}, \tag{5.4.3}$$

where d is the dimensionality of θ.

Proof: Let $0 \leq ||\lambda_{\alpha 1}|| \leq \cdots \leq ||\lambda_{\alpha d}|| \equiv \rho_\alpha \leq 1$ be the ordered eigenvalue norms (in complex space) of DM^{ECM_α} ($\rho_\alpha \leq 1$ whenever ECM $_\alpha$ converges). Using the fact that the determinant is the product of the eigenvalues, we have

$$(1 - \rho_\alpha)^d \leq \prod_{i=1}^{d} ||1 - \lambda_{\alpha i}|| = |I - DM^{ECM_\alpha}| = |I - DM^{ECM_\beta}| \leq 1 - \rho_\beta,$$

where the first inequality follows from $(1 - \rho_\alpha) \leq ||1 - \lambda_{\alpha i}||$ for $i \leq d$.

■

Condition (5.4.2) is satisfied whenever all the eigenvalues of DM^{ECM_β} are real and positive, a condition that holds for any EM algorithm. Unlike DM^{EM}, however, it is possible for DM^{ECM} to have imaginary or negative eigenvalues. But when this happens, our simulations show that the imaginary part of complex eigenvalues or negative eigenvalues tends to be small in magnitude. We thus expect that condition (5.4.2) can eventually be removed or proved to be true under simpler conditions.

Although the bound in the corollary is very crude, it is nontrivial even for quite large values of d (e.g., $d \geq 20$). To see this we use (5.2.3) to put the corollary in terms of the number of iterations required by the fastest and slowest algorithms

that result form permuting CM-steps. Specifically, suppose $N_{\max}$ and $N_{\min}$ are the number of iterations required by the slowest and fastest algorithms respectively, then

$$\frac{N_{\min}}{N_{\max}} \geq \frac{\log(\rho_{\max})}{\log(1 - \sqrt[d]{1 - \rho_{\max}})}, \qquad (\mathbf{5.4.4})$$

where $\rho_{\max}$ is the root convergence factor of the slowest algorithm. A plot of the right side of (5.4.4) for several values of d appears in Figure 5.5. This represents the maximum improvement that we could see by permuting CM-steps as a function of the slowest rate of convergence. Notice that the slower the algorithm (large ρ) the more improvement we can hope for, which is consistent with our intuition. We see that the crude bound provided by the corollary eliminates a nontrivial part of the impossible values of $N_{\min}/N_{\max}$ (i.e, the area below the curves in Figure 5.5).

5.4.2. Empirical Results

Recalling that (5.4.4) is based on the theoretical approximation (5.2.3), we again turn our attention to the simulations described in Section 5.3 regarding the actual number of steps required. The simulations use a three CM-step ECM algorithm and falls under Theorem 5.2. The six possible ECM algorithms were run on each of the data sets. The quantity $N_{\min}/N_{\max}$ was calculated along with the empirical root convergence factor at convergence of the slowest algorithm, $\hat{\rho}_t$. Figure 5.6 shows a plot of these for the three simulations along with results from a fourth simulation run with $n = 200$, $n_c = 140$, and $n_i = 20$ for each i, which was added to include smaller values of ρ (less missing information). The solid line is the right side of (5.4.4) with $d = 6$. Given that we have seen substantial discrepancy between

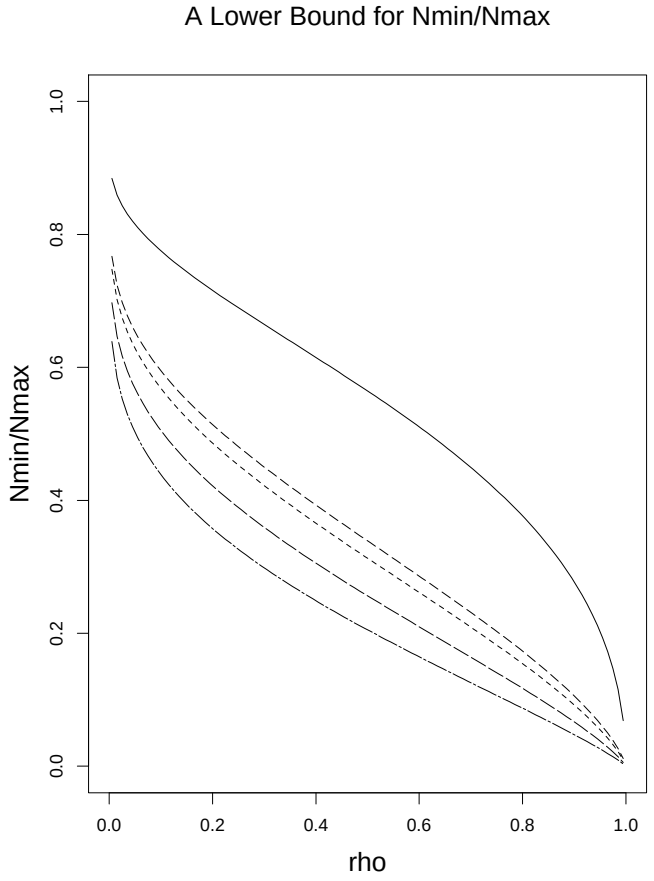

Figure 5.5. Bounding $N_{\min}/N_{\max}$. This is a plot on the bound givin in (5.4.4) for $d = 2, 3, 5, 10,$ and 20 (from top to bottom).

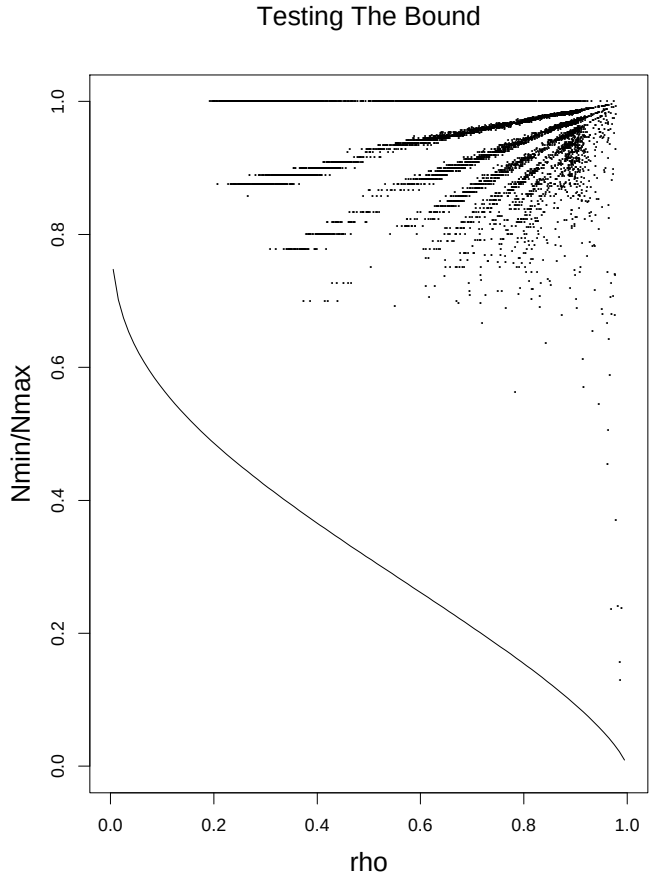

Figure 5.6. Testing the bound on $N_{\min}/N_{\max}$. We plot the 8000 simulated values of $\hat{\rho}$ versus the relative increase in efficiency due to permuting CM-steps compared with the bound given in (5.4.4) for $d = 6$.

theoretical approximations and simulation studies, it is somewhat remarkable that (5.4.4) holds for all the simulated data sets. This probably is because the bound is crude and therefore conservative, but the plot also indicates that it is approachable by a few actual cases. We also notice that except for a few extreme cases which converge very slowly, there seems to be little empirical evidence for substantial increased relative improvement with ρ.

5.5. Factors Affecting Relative Gain

In Figure 5.4, we see that the relative reduction in the number of steps required by choosing the optimal order is typically within 20%. But there is a non-trivial portion of cases where the relative gain is more substantial. In practice, we would like to know what factors in the data and of the model will make large improvements likely. In this section we will discuss three possibilities: the number of CM-steps, the relative amount of incomplete data, and the sparseness of the data relative to the number of model parameters.

5.5.1. The effect of the number of CM-steps

In order to evaluate the impact of the number of CM-steps, we ran a simulation on a $2 \times 2 \times 2 \times 2$ table. This simulation was designed to be comparable with simulation 1 (see Table 5.1) and is described in the second row of Table 5.3 as simulation 5. As in simulation 1 there were n_1, n_2 and n_3 observations which were not classified

Sim.	n	n_c	n_1	n_2	n_3	$100 \times \iota_s$	ι_i
1	100	25	25	25	25	6.00	0.75
5 †	233	59	58	58	58	6.00	0.75
6	40	16	8	8	8	15.00	0.60
7	75	30	15	15	15	8.00	0.60
8	150	60	30	30	30	4.00	0.60
9	500	200	100	100	100	1.20	0.60
10	10000	4000	2000	2000	2000	0.06	0.60
11	100	16	28	28	28	6.00	0.84
12	100	31	23	23	23	6.00	0.69
13	100	46	18	18	18	6.00	0.54
14	100	61	13	13	13	6.00	0.39
15	100	76	8	8	8	6.00	0.24
16	50	10	30	10	0	12.00	0.80

† Four CM step algorithm for a $2 \times 2 \times 2 \times 2$ contingency table

Table 5.3. Simulation sample sizes for simulations 5 – 16. This table records the total number of observations n, the number of observations that are completely classified n_c, the number n_i which were classified only according to margin i, the index of sparseness ι_s, and the index of the amount of incomplete data ι_i.

by two of the categorical variables, this time resulting in three 2×2 marginal tables. In order to keep simulation 5 comparable with simulation 1, the index of the amount of incomplete data (i.e. $\iota_i \equiv 1 - n_c/n$) and the index of sparseness (i.e. $\iota_s = (\text{number of parameters})/n$) were held constant as described in the first two rows of Table 5.3. The data was simulated by the method analogous to that described in Section 5.3, and we fit the model with only main effects and two-way interactions. This model requires six CM-steps and resulted in $6! = 720$ possible algorithms, all of which were run for 1000 simulated data sets. Again, for each data set the number of iterations required by the fastest and the slowest algorithms were recorded, $N_{\min}$ and $N_{\max}$ respectively. The quantity $N_{\min}/N_{\max}$ is the

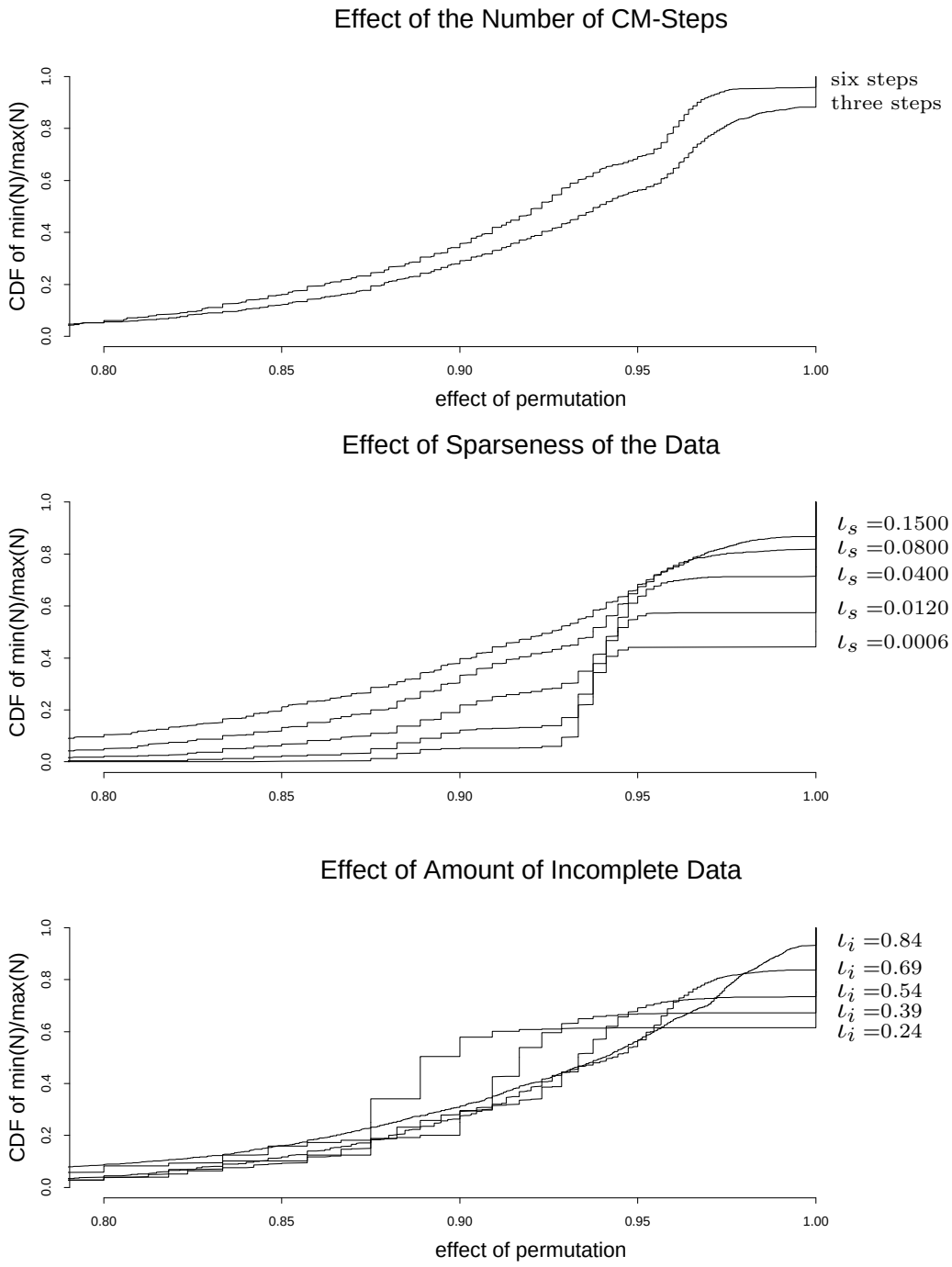

Figure 5.7. CDFs of the number of iterations required for convergence for the simulations described in Table 5.3. The CDFs demonstrate the effect of the number of CM steps and sparseness of the table; and the lack of effect of the missing data structure.

complement of the maximum percent increase in efficiency due to permutation of CM-steps and its cumulative density function (CDF) appears as the top line in the top panel of Figure 5.7. This can be compared with the bottom line which is the CDF for the same statistic from simulation 1. Since the CDF of simulation 1 is lower than that of simulation 5 we see that the six step algorithm tends to produce lower values of $N_{\min}/N_{\max}$, and thus permutations are more important in the six step algorithm. This, of course, confirms our intuition that the more CM-steps we have, the more possible orders there are, and thus the more discrepant the minimum and maximum number of steps required should be.

5.5.2. The effect of sparseness

To evaluate the effect of the amount of incomplete data and sparseness, we return to the $2 \times 2 \times 2$ table described in Section 5.3. Ten additional simulations were run and are described in Table 5.3. Simulations 6 – 10 are designed to look at the effect of sparseness and thus the amount of incomplete data were fixed at $\iota_i = 60\%$ for each of them. The CDF of $N_{\min}/N_{\max}$ for each of these simulations appears in the second panel of Figure 5.7; simulation 6 is on the top and simulation 10 is on the bottom. The effect of sparseness is clear – permutations make more difference in tables with less data. In particular, in tables with several zero cells permutations are especially important. In addition to the $2 \times 2 \times 2$ tables, in simulation 5 which used a six CM-step algorithm, in 0.8% of the $2 \times 2 \times 2 \times 2$ tables generated, a relative improvement in efficiency of over 90% was observed (e.g. optimal order is more than 10 times faster than the slowest order). All of these tables had several

zero cells. This finding suggests that it is more important to study the issue of ordering with sparse tables, where the gain (or loss) can be very substantial. The reason that sparseness is important is somewhat elusive. One possible explanation, however, stems from the fact that each CM-step computes MLEs conditional on a function of the parameters and the data. When there are more data relative to the number of parameters, the data become relatively more important in this computation. The function of the parameters becomes less important and thus the order in which we condition on different functions of the parameters matters less.

5.5.3. The effect of the amount of incomplete data

Simulations 11 – 15 investigate the effect of the amount of incomplete data and the sparseness was fixed at $\iota_s = 0.06$. The corresponding CDFs of $N_{\min}/N_{\max}$ appear in the final panel of Figure 5.7. When there are more incomplete data, the CDFs tend to be smoother since they require more steps to converge and thus the integer division in $N_{\min}/N_{\max}$ results in less of a step function. On the other hand, the *relative* decrease in steps required does not seem to change as ι_i increases. It should be noted, however, that when there is more missing information the ECM algorithm is slower so that even with the same relative increase in efficiency, the *absolute* increase in efficiency (i.e. absolute time saved) will increase with ι_i .

Another way to look at the effect of incomplete data is to look for a relationship between the missing-data structure and the specific order of CM-steps. For example, in a $2\times2\times2$ table, when the no three-way interaction model is fit, each of the three CM-steps estimates the main effects and the two-way interaction in one of

the three 2×2 marginal tables. Depending on the incomplete-data structure, these marginal tables may have more or less incomplete data. To be specific, consider simulation 16 of Table 5.3, where 60% of the data are classified only according to the first margin. Thus, when we add across this margin these data will be completely unclassified. Since $n_3 = 0$, however, when we add across the third margin all of the data will be at least partially classified. It therefore may be efficient in this simulation to first run the CM-step that operates on the marginal table which sums over the third margin, that is, the first CM-step is to fit the 2×2 table corresponding to the first two factors. This strategy has been a rule of thumb in practice, and seems to be inspired by the common technique of factoring the likelihood when some marginal densities have no missing data. For example, an application of ECM to bivariate normal data (Section 4.4.5) illustrates the computational advantage of estimating first those parameters with no missing information. In the context of simulation 16, however, there seems to be little benefit in first running the CM-step with the most data (i.e., the step corresponding to summing over the third factor). Figure 5.8 compares the log of the number of iterations required for each of the six possible orderings. Each of the smaller plots compares two orderings and each point represents one of the 1000 data sets. There is no indication that any one ordering tends to be more efficient. This was confirmed by several other simulations. The missing-data structure does not seem to help us identify the optimal order of CM-steps, at least not as much as conventional wisdom would have predicted.

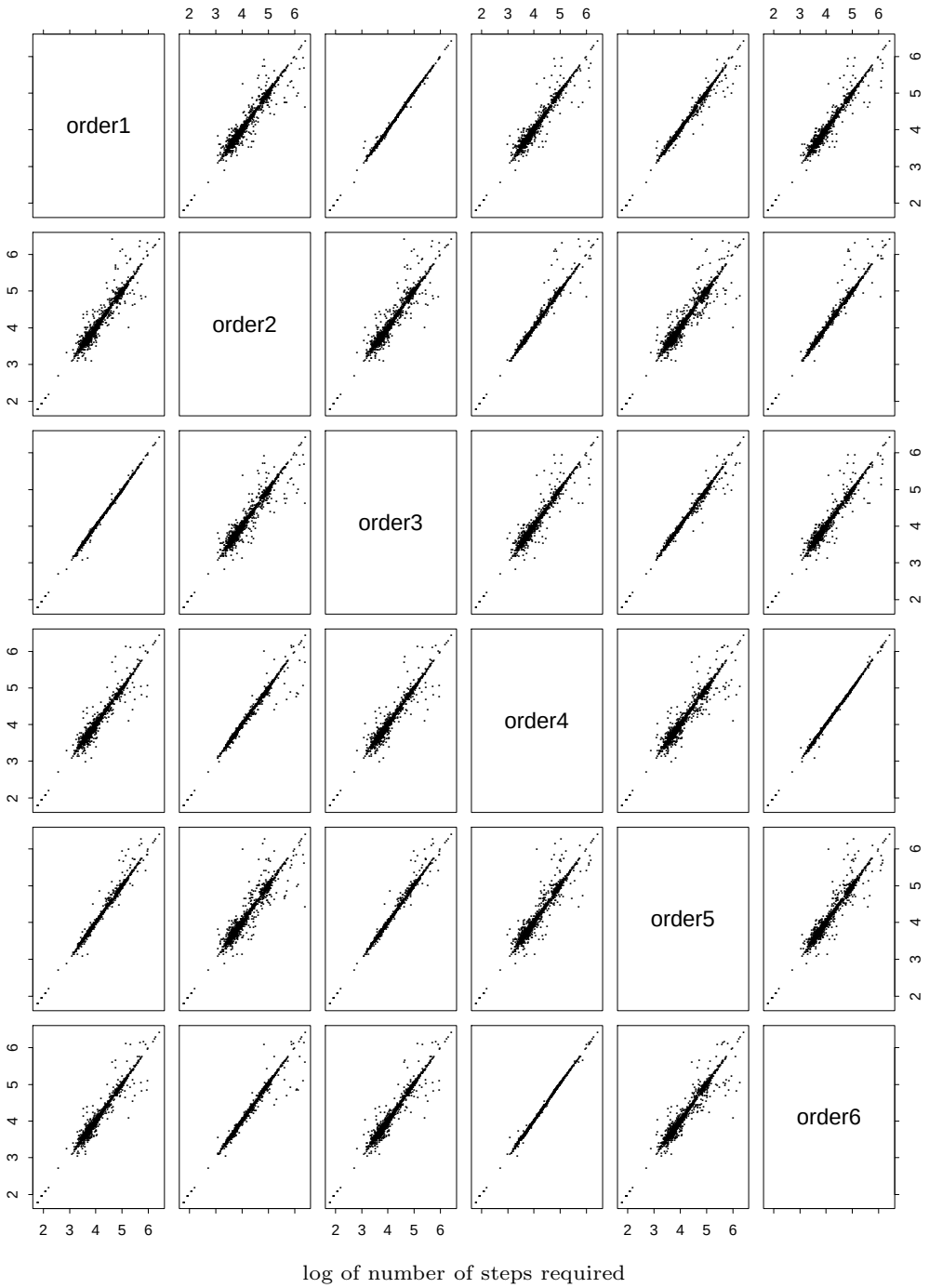

Figure 5.8. This plot compares the log of the number of steps required for each pair of the six possible ordering of CM steps. Notice how closely orderings 1 and 3 and orderings 4 and 6 coincide.

5.5.4. Summary of factors affecting relative gain

To summarize the results we can look at the ECM algorithm as a composite of the EM algorithm which works on the missing data and the CM algorithm which introduces model reduction via conditional maximization. More missing data slow down the EM or ECM algorithms, as is evident by the relative smoothness of the CDFs in the bottom plot in Figure 5.7, but has little effect on how the CM-steps interact. Whether or not there are missing data it is the sparseness of the data defined by ι_s , and the number of CM-steps that are important in gains due to CM-step permutation. Since neither of these are characteristics of the missing data, it may suffice to look only at the characteristics of the CM algorithm in order to understand how to improve ECM through CM-step permutation. In the next section we will look at a strategy that has been suggested in a stochastic version of the CM algorithm known as stochastic relaxation.

5.6. The Cycled and Random ECM Algorithms

5.6.1. The cycled ECM algorithm

If changing the order of CM-steps greatly affects the rate of convergence of ECM, we could lose efficiency by making the wrong choice of how to order the CM-steps. Lacking a method for choosing a good order, we might hope to decrease the risk of a badly inefficient algorithm by somehow averaging the orderings. One strategy a practitioner might employ is the "cycled" ECM algorithm described below.

Given an ECM algorithm with S CM-steps, let ECM $_{\alpha_i}$ ($i = 1, \ldots, S!$) be the algorithms resulting from all the possible permutations of the CM-steps. The cycled ECM algorithm runs one iteration from each of these algorithms in an arbitrary order, and then continues to cycle through them until convergence. As will be described in Section 5.7, this can create instability in the step size $||\theta^{(t)} - \theta^{(t-1)}||$, and care must be taken when evaluating convergence of the algorithm. In the simulation, we ran the algorithms until the difference between consecutive loglikelihood values, $L(\theta^{(t+1)}|Y_{\text{obs}}) - L(\theta^{(t)}|Y_{\text{obs}})$, was within a pre-specified threshold. Since this requires the evaluation of the actual likelihood, it may not always be feasible in practice, although it is always desirable, since it allows us to be sure the likelihood is increasing at each iteration, a feature of ECM when it is implemented correctly.

5.6.2. Empirical evaluation of cycled ECM

The rationale behind the cycled ECM algorithm is the hope that although it will not be as fast as the fastest ordering of CM-steps, it also should not be as slow as the slowest. It turns out that neither of these is true. In the simulations described in Table 5.1, in addition to the six ECM algorithms, we ran the cycled ECM algorithm that incorporated the six ECMs. There were cases where cycled ECM was the fastest, but also cases where cycled ECM was the slowest. A useful comparison is to compare cycled ECM with a randomly selected ECM algorithm. That is, on the outset of running the ECM algorithm, an order for the CM-steps is chosen at random and fixed until convergence. Let the number of iterations required for convergence of the resulting ECM be N_f. This is compared with N_c, the number

Sim.	$N_f > N_c$	$N_f = N_c$	$N_f < N_c$
1	0.340	0.141	0.512
2	0.447	0.095	0.459
3	0.325	0.207	0.468

Table 5.4. N_c versus N_f. The number of iterations required for convergence for the cycled ECM algorithm (N_c) and ECM with CM-steps in a randomly chosen fixed order (N_f) for the 6000 simulated data sets described in Table 5.1.

of iterations required for cycled ECM. For a fair comparison, we define an iteration to be one E-step followed by three CM-steps for both algorithms (regardless of the order of CM-steps). The results for the three simulations in Table 5.1 appear in Table 5.4, and make it clear that on average cycled ECM offers no advantage over haphazard selection of a fixed order of CM-steps, at least in this example.

5.6.3. The random ECM algorithm

In what has been described above, the order in which we cycle through the ECM algorithms is chosen in advance. Instead of doing this, however, we could randomly select an order at each iteration. The rational for this strategy stems from Amit and Grenader's (1991) recommendation in the context of stochastic relaxation (or Gibbs Sampler), a procedure useful when it is difficult or impossible to draw from the joint density $\mathcal{L}(\theta_1, \ldots, \theta_S)$, but it is relatively easy to draw from all conditional densities $\mathcal{L}(\theta_s|\theta_{s'}; s' \neq s)$ $(s = 1, \ldots S)$. This situation is very similar to situations where ECM is useful: the augmented-data joint MLE for $(\theta_1, \ldots \theta_S)$ is hard to calculate, but the augmented-data conditional MLE of θ_s given $\{\theta_{s'}; s' \neq s\}$ is easy to obtain for all s. With stochastic relaxation, given the current draws $(\theta_1^{(t)}, \ldots, \theta_S^{(t)}), \theta_1^{(t+1)}$ is drawn from $\mathcal{L}(\theta_1|\theta_2^{(t)}, \ldots, \theta_S^{(t)})$, and so on until $\theta_S^{(t+1)}$ is

Sim.	$N_f > N_r$	$N_f = N_r$	$N_f < N_r$
1	0.330	0.144	0.527
2	0.445	0.095	0.460
3	0.323	0.218	0.460

Table 5.5. N_r versus N_f. The number of iterations required for convergence for the random ECM algorithm (N_r) and ECM with CM-steps in a randomly chosen fixed order (N_f) for the 6000 simulated data sets described in Table 5.1.

drawn from $\mathcal{L}(\theta_S|\theta_1^{(t)}, \ldots, \theta_{S-1}^{(t)})$. Replacing the conditional draws by conditional maximizations, the CM-steps of ECM mimic exactly the same process. In this sense, ECM can be regarded as a deterministic version of stochastic relaxation, as discussed in Meng and Rubin (1992). Amit and Grenander (1991) found bounds on the convergence rates of stochastic relaxation for both deterministic periodic and random orderings. Their bound for random orderings was lower and they therefore recommended that in the absence of prior knowledge of the particular form of the iteration, the random ordering be used. Given the similarity between ECM and stochastic relaxation, we hypothesized that random orderings would outperform deterministic orderings with the ECM algorithm as well.

To check our hypothesis we again turned to the simulations described in Table 5.1. We compared the number of iterations required by the algorithm that randomly selects an order at each step, N_r, and the algorithm that randomly selects an order on the outset, N_f. The comparison appears in Table 5.5 and shows that cycled and random order algorithms are essentially indistinguishable, and thus we have not obtained evidence for the advantage of using these more sophisticated "averaging" strategies instead of simply fixing an arbitrary ordering at the outset.

5.7. Discussion

5.7.1. Convergence Criteria

In this section, we will show how the standard "step length" criterion can be quite misleading and discuss the sensitivity of the results of the previous sections to the convergence criterion. In all the simulations except those of Section 5.6, we used the standard step length criterion: $||\theta^{(t)} - \theta^{(t-1)}|| \leq \epsilon$, which was useful for our purposes both because of its popularity and because it underlies (5.2.3) which relates the number of steps required for convergence to the spectral radius. In the context of cycled ECM, however, this criterion can lead to difficulties. If each iteration of cycled ECM is defined as one E-step followed by S CM-steps, then the resulting sequence $\{\theta^{(t)} : t \geq 0\}$ is not a simple linear iteration even at convergence since the mapping that maps $\theta^{(t)}$ to $\theta^{(t+1)}$ changes with t. The difficulty with this is demonstrated in Figure 5.9, which is a representation of the mapping induced on a subspace of Θ by three ECM algorithms each with a fixed ordering (the smooth curves) and by the cycled ECM algorithm that combines them (the jagged curve). The three ECM algorithms are listed in the first row of Table 5.2. This is again an example of fitting a log-linear model to a partially classified $2 \times 2 \times 2$ contingency table. The asterisks represent the sequences and show that the cycled ECM algorithm tends to have larger steps. In fact, using the standard convergence criterion, $||\theta^{(t+1)} - \theta^{(t)}|| \leq \epsilon$, this cycled ECM algorithm took 50 times longer to converge than any of the three ECM algorithms. This is in spite of the fact that all

the algorithms increased the loglikelihood at about the same rate. (Cycled ECM is the solid line in Figure 5.10.) Since the loglikelihood is increased at each iteration an alternative convergence criterion is $L(\theta^{(t+1)}|Y_{\text{obs}}) - L(\theta^{(t)}|Y_{\text{obs}}) < \epsilon$, which is more desirable in the context of maximum likelihood estimation as described in Section 5.6.

This brings up the question of how sensitive the results of the simulations are to the convergence criterion. Figure 5.11 shows the values on $N_{\min}/N_{\max}$ (as defined in Section 5.4) for the simulations in Table 5.1 using the step length criterion with $\epsilon = 10^{-10}$ and loglikelihood criterion with $\epsilon = 10^{-8}$. The loglikelihood criterion leads to more smaller values of $N_{\min}/N_{\max}$ than does the standard length criterion. Thus, CM-step permutations appear to be more important when the loglikelihood criterion is used. (The comparisons here are not strictly "fair" because of the choices of the threshold ϵ. However, it is difficult to decide what is fair here, and the impact of ϵ on the relative value $N_{\min}/N_{\max}$ should be small as long as ϵ is not too big.)

Certainly, the convergence criterion plays an important role in both simulation and practice and thus must be chosen carefully. The fact that the step length criterion failed in cycled ECM shows that the practitioner must be careful when evaluating convergence of algorithms that change even in subtle ways at each iteration. Whenever possible, we recommend the use of the likelihood criterion, in addition to the step length criterion.

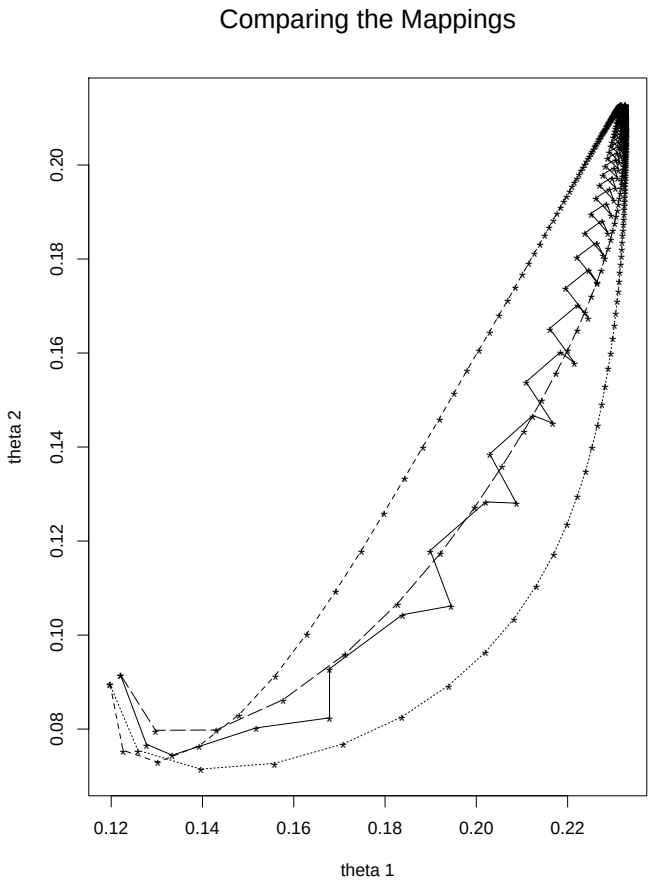

Figure 5.9. The mapping induced by the cycled ECM algorithm. The mapping induced on the parameter space (cell probabilities) by three ECM algorithms (dotted lines) and the composite cycled algorithm (solid line). Notice the jagged step pattern of the cycled ECM algorithm.

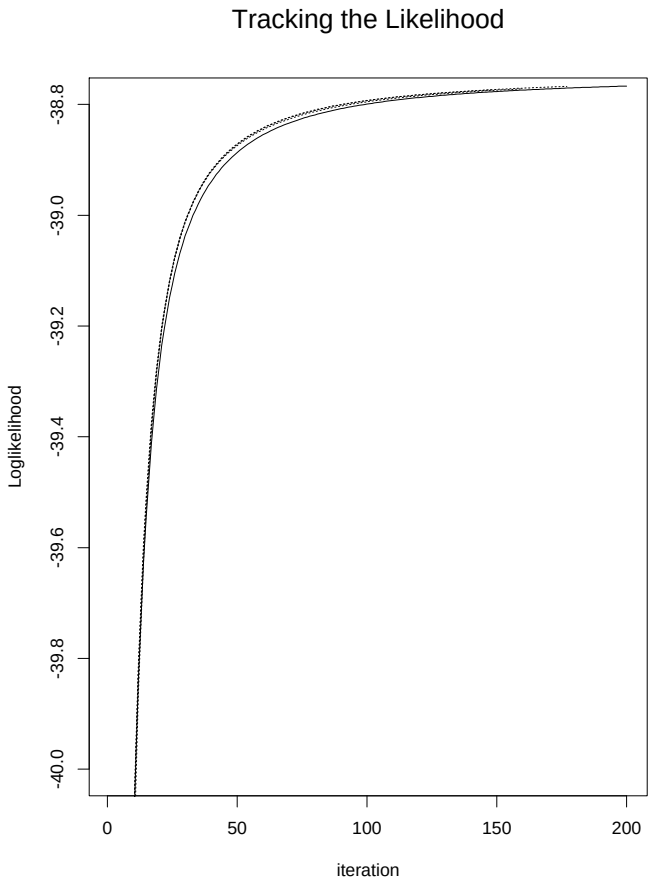

Figure 5.10. Tracking the likelihood with the cycled ECM algorithm. The figure shows how the four algorithms mapped in Figure 5.9 increase the loglikelihood. The fixed order ECM algorithms (upper dashed lines) are indistinguishable and increase the loglikelihood somewhat faster than the cycled algorithm (lower solid line).

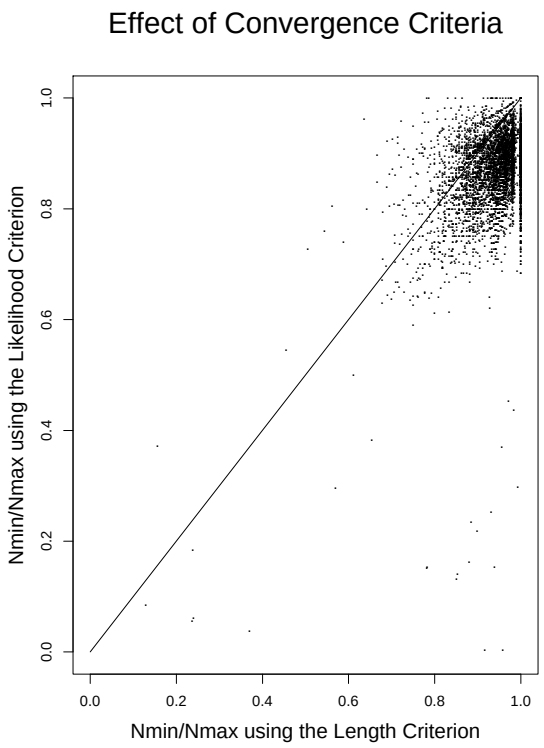

Figure 5.11. Compareing the loglikelihood and step length convergence criteria. This is a plot of $N_{\min}/N_{\max}$ using both criteria for the 6000 simulations described in Table 5.1. Notice that the effect of CM-step permutation tends to be larger with the loglikelihood criterion.

5.7.2. Stratagies for ordering CM-steps

The simulations presented in Sections 5.3 and 5.4 show that the effect of permuting CM-steps on the number of iterations required for convergence can be much more substantial than the theoretical results predict. This finding is rather informative since most studies of the convergence of EM type algorithms have been conducted purely with regards to the asymptotic character of the iteration. The result, however, is somewhat discouraging in light of how evasive exact analysis of the iteration can be.

When considering the problem of ordering, perhaps the question of most interest is how large the effect of ordering can be. We have seen that an increase

in efficiency of 20% is not uncommon in the $2 \times 2 \times 2$ table. Other simulations suggest that in problems with more CM-steps the effect tends to be larger. In the contingency table setting, the sparseness of the table is also important. In tables with less data, the effect of CM-step permutation can be greater. Effective strategies for maximizing efficiency continue to be elusive. Neither relative amounts of missing data nor characteristics of the subspaces onto which the CM-steps project (i.e., Alternating Projection Algorithm, Kayalar and Weinert, 1988) have been found to be useful in determining a general technique for choosing an efficient ordering. Based on our limited simulations, a user who is faced with a sparse table with many CM-steps is perhaps best off randomly selecting the order of CM-steps at the outset of running the algorithm. In general, when one is unsure if more sophisticated strategies will pay off, we recommend randomly fixing an order of CM-steps before implementing the ECM algorithm.

Finally, we emphasize that our general philosophy behind investigating the effect of permuting CM-steps is to see if there is a "free and better lunch," not a "better but costly lunch" in terms of human and computational effort. Given the diversity of applications of EM-type algorithms, it seems impossible to find an "optimal" order-choosing rule that will be universally applicable. Even if such a rule could be found, it has no practical value unless the savings it provides outweighs the cost of implementing it. On the other hand, a practitioner may be interested in knowing about strategies that will lead to relatively efficient algorithms in common implementations of EM-type algorithms (e.g, ECM applied to contingency tables) especially when the algorithm is slow to converge, and in what situations the issue

of ordering is not worth consideration because it is unlikely to have an appreciable impact on the problem at hand. In either case, the practitioner is informed by a general investigation that helps to answer these questions.

Chapter 6

The Global Rate of EM as an Inferential Tool: Finite Mixtures

6.1. Finite Mixture Distributions

Finite mixtures appear in many areas of the statistical literature. The density of the general finite mixture distribution is the weighted average of the k densities (subpopulations) $f_i(y|\beta_i)$ and can be formulated as

$$f(y|\theta_k) = \sum_{i=1}^{k} \alpha_i f_i(y|\beta_i), \qquad \textbf{(6.1.1)}$$

where $\sum \alpha_i = 1$ and $\theta_k = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k)$. Attempts to estimate the parameters in (6.1.1) date back at least to Pearson's (1894) method of moments solution to a mixture of two univariate normals with different means and variances. The algebra was formidable indeed, the first step of the solution requiring a negative root of a "nonic" equation. Since then graphical, likelihood, Bayesian, and minimum density distance solutions have been proposed (see Titterington, Smith, and Markov, 1985 for a nice review). Even when k is known, likelihood estimation has generally been hindered by the fact that the loglikelihood does not take the

familiar additive form. A simple yet general computational strategy, however, is to apply the EM algorithm, treating the subpopulation memberships (i.e., mixture indicator) as missing data.

The mixture problem is further complicated when the number of subpopulations k is unknown, and thus needs to be estimated. The two facts that k is discrete and that $\sup_{\theta_k \in \Theta_k} f(y|\theta_k)$ is nondecreasing in k make estimation of k a nontrivial task. There is a diverse and creative literature of attempts to solve this rather challenging estimation and the related testing problems. Noticing that $f(y|\theta_{k+1})$ is a just $f(y|\theta_k)$ with $\alpha_{k+1} = 0$ (or $f_{k+1}(y|\beta_{k+1}) = f_k(y|\beta_k)$), Wolfe (1970) proposed using the likelihood ratio test with its standard asymptotic distribution. It was pointed out later, however, that the fact that the constrained space is on the boarder of the parameter space destroys the standard asymptotic result. Others have suggested methods based on the chi-square distance from the empirical CDF (Henna, 1985), modified likelihood ratio tests (Aitkin and Rubin, 1985), Monte Carlo estimation (Aitkin, Anderson and Hinde, 1981), graphical techniques (c.f., Titterington, Smith and Makov , 1985), and order statistics (Maine, Boullion, and Rizzuto, 1991).

One particularly interesting approach was proposed by Windham and Cutler (1992), in which the rate of convergence of EM is used to estimate k. As we have seen, the larger the augmented information (relative to the observed information), the slower the EM algorithm converges. In the mixture problem, EM is run conditional on an assumed k and the augmented data is $\{(y_i, z_i), i = 1, \ldots n\}$, where z_i indicates the subpopulation membership of observation y_i. Windham and Cutler

(1992) argued that if the assumed k is too large or too small compared to the underlying true k, the subpopulation memberships are "ill-defined" making the algorithm slow. Thus, after running several EM algorithms with different values of $k(\geq 2)$, they propose to estimate k as the number of subpopulations that results in the fastest EM algorithm. This idea suggests that the augmented-data loglikelihood contains information for the parameter k beyond what is contained in the observed-data loglikelihood.

Windham and Cutler's argument for this method was largely heuristic and their evidence came from a single example. Nevertheless, the idea is not only novel but also if successful has great potential in EM application outside of finite mixture models. In this chapter, we will continue to explore the usefulness as well as the limitations of this method. In Section 6.2, we introduce the necessary details of the EM algorithm used to fit finite mixture models and its rate of convergence and in Section 6.3 we briefly review the details of Windham and Cutler (1992). In Section 6.4, we show that Windham and Cutler's approach can be viewed as a minimax estimator. We then discuss the large-sample behavior of the estimator, conducting an empirical study (Section 6.5) and presenting a more theoretical argument (Section 6.6), both of which indicate that the estimator can be inconsistent and tends to underestimate the true k. Finally, Section 6.7 proves several propositions presented in Section 6.6 and presents an outline of a more thorough theoretical investigation.

6.2. Using EM to Fit Finite Mixture Models

As described in Dempster, Laird, and Rubin (1977), to implement the EM algorithm in the context of finite mixture analysis, we augment each of the observed-data points, y_i with its (unknown) subpopulation membership, z_i, where z_{ij} is the indicator function of whether y_i belongs to subpopulation j. We will label the observed values $Y_{\text{obs}} = \{y_1, \ldots, y_n\}$ and the augmented data $Y_{\text{aug}} = \{(y_i, z_i), i = 1, \ldots, n\}$. Given a *working* model $f(y|\theta_k)$ we can write the expected augmented-data loglikelihood

$$Q(\theta_k|\theta_k^{(t)}) = \int L(\theta_k|Y_{\text{aug}}) f(Y_{\text{mis}}|Y_{\text{obs}}, \theta_k^{(t)}) \mathrm{d}Y_{\text{mis}}. \qquad (\mathbf{6.2.1})$$

Since this function is linear in $\{z_{ij}, i = 1, \ldots, n\}$, the E-step amounts to computing

$$\mathsf{E}[z_{ij}|Y_{\text{obs}}, \theta_k^{(t)}] = \frac{\alpha_j^{(t)} f_j(y_i|\beta_j^{(t)})}{\sum_{l=1}^{k} \alpha_l^{(t)} f_l(y_i|\beta_l^{(t)})}, \quad j = 1, \ldots, k, i = 1, \ldots, n, \qquad (\mathbf{6.2.2})$$

which is simply $\mathsf{P}(z_{ij} = 1|\theta_k^{(t)}, y_i)$. The M-step then maximizes $Q(\theta_k|\theta_k^{(t)})$, a computational task that is the same as maximizing the augmented-data loglikelihood $L(\theta_k|Y_{\text{aug}})$, with z_{ij} replaced by the probability found in (6.2.2).

In the context of the mixture problem, both the matrix rate DM^{EM} and the global rate r will depend on the working model $f(y|\theta_k)$ and thus on k, as well as on the data and thus the true model $f(y|\theta_{k_0})$, where k_0 may be different from k (here we assume that the true model is indeed a finite mixture). Since k and k_0 are of primary interest here, we will write the matrix rate $DM_n^{EM}(k|k_0)$ and the global rate (i.e., the largest eigenvalue of $DM_n^{EM}(k|k_0)$) $r_n(k|k_0)$ [†], where n indexes the

[†] The subscript on r will always be the sample size in this chapter and should not be confused with the subscript on r_t as defined in (1.6.9), which indicates the iteration number in the computational formula for r.

sample size (i.e., the number of observations in Y_{obs}). For computational purposes, we will use $\hat{r}_n(k|k_0) = \lim_{t\to\infty} ||\theta_k^{(t)} - \theta_k^{(t-1)}||/||\theta_k^{(t-1)} - \theta_k^{(t-2)}||$.

6.3. The Windham and Cutler Approach

The fraction of observed information or information ratio matrix $I - DM_n^{EM}(k|k_0) = I_{\text{obs}} I_{\text{aug}}^{-1}$ is presented in Windham and Cutler (1992) as a measure of "the ability of the data to distinguish the component densities [of the mixture]." Specifically, when the ratio is close to the identity, I_{obs} will be large relative to I_{aug} and Y_{mis}, the subpopulation memberships, contains little more information than the observed-data themselves. For example, letting $\phi(\mu, \sigma)$ be the normal density with mean μ and variance σ^2, if we have 1000 observations from a mixture of $\phi(0, 1)$ and $\phi(20, 1)$, even without Y_{mis}, there is little question as to the subpopulation memberships. In general, directly using $DM_n^{EM}(k|k_0)$ can be complicated since it can be of rather high dimension; for example, a mixture of two trivariate normals results in a 19×19 rate matrix. As we have seen, however, the largest eigenvalue of $DM_n^{EM}(k|k_0)$ corresponds to the global rate of convergence of the algorithm and is a good summary of the matrix. With this in mind, Windham and Cutler (1992) propose fitting mixtures with several different values of $k \geq 2$, and estimating k with $\hat{k}_r$, the number of subpopulations that results in the fastest EM algorithm, as calculated with the $\hat{r}_n(k|k_0)$:

$$\hat{k}_r \equiv \text{argmin}_{k\geq 2} \{\hat{r}_n(k|k_0)\}. \qquad (\textbf{6.3.1})$$

The restriction $k \geq 2$ is needed because when $k = 1$, there is no missing data and $DM^{EM}(1|k_0) = 0$. Thus, we will only consider nontrivial mixtures. It should also be mentioned that Windham and Cutler define $\hat{k}_r$ as the k that maximizes $1 - \hat{r}_n(k|k_0)$, which they call the minimum information ratio (MIR).

Windham and Cutler's basic justification for this procedure, in addition to the heuristic argument presented above, was the application to a mixture of three spherical bivariate normals with known and equal standard deviation. When the standard deviation was small relative to difference in means, it is not surprising that the method worked well (almost any method would work well). When the standard deviation was relatively large, however, their tables reveal that $\hat{k}_r$ picked the true value of k only about half the time and underestimated almost every other time. As we will see, this pattern continues in other examples.

To fully understand and explore this idea, we conducted both theoretical and simulation studies, with a focus on constructing possibly better estimating procedures.

6.4. The Rate Function and the Minimaxity of $\hat{k}_r$

It is clear from (1.6.2) – (1.6.5) that $r_n(k|k_0)$ is attempting to measure differences between the observed-data loglikelihood function, $L(\theta_k|Y_{\text{obs}})$ and the projection of the augmented-data loglikelihood onto the observed-data space, i.e., the conditional expectation of the augmented-data loglikelihood, $Q(\theta_k|\theta_k^\star)$, where $\theta_k^\star$ is the

maximum likelihood estimate of θ_k. It remains unclear, however, what specific properties it is measuring, or if what it is measuring is an appropriate basis for estimating k_0. In this section we introduce a more general and direct measure of the difference between $L(\theta_k|Y_{\text{obs}})$ and $Q(\theta_k|\theta_k^\star)$, inspired by $r_n(k|k_0)$. Specifically, we define the rate function as

$$R(\theta|Y_{\text{obs}}) = 1 - \frac{L(\theta^\star|Y_{\text{obs}}) - L(\theta|Y_{\text{obs}})}{Q(\theta^\star|\theta^\star) - Q(\theta|\theta^\star)}, \qquad (\textbf{6.4.1})$$

where $\theta^\star$ is the limit of an EM sequence. (We have suppressed the subscript k because the theory is general.) In order to gain an intuition for the numerical properties of this function, we generated 10 000 observations from $Y_{\text{obs}} \sim \frac{1}{2}\phi(0,1) + \frac{1}{2}\phi(3,1)$, used EM to fit the model $Y_{\text{obs}} \sim \alpha\phi(\mu_1,\sigma) + (1-\alpha)\phi(\mu_2,\sigma)$ and plotted (6.4.1) as a function of each of the four parameters fixing the other three at their MLE (see Figure 6.1). The wild fluctuations of $R(\theta|Y_{\text{obs}})$ near the $\theta_k = \theta_k^\star$ are due to numerical inaccuracy as the denominator in (6.4.1) approaches zero. The actual plot near $\theta^\star$ can be interpolated as a smooth function. We would like to compare $R(\theta|Y_{\text{obs}})$ with the rate of convergence of EM, which is plotted as the horizontal dashed line on each of the plots. From the plots, we see $R(\theta|Y_{\text{obs}})$ is always in the unit interval, that the $\lim_{\theta\to\theta^\star} R(\theta|Y_{\text{obs}})$ does not exist, and that each of the limits along the axes of Θ are different but are less than the rate of convergence of EM. Theorem 6.1 clarifies these properties.

Theorem 6.1: If $\theta^\star$ is the global maximizer of $L(\theta|Y_{\text{obs}})$ then the function $R(\theta|Y_{\text{obs}})$ defined in (6.4.1) has the following property

(1) $0 \le R(\theta|Y_{\text{obs}}) \le 1$ for all $\theta \in \Theta$.

Furthermore, if $\nabla L(\theta|Y_{\text{obs}})\Big|_{\theta=\theta^\star} = \nabla Q(\theta|\theta^\star)\Big|_{\theta=\theta^\star} = 0$, then

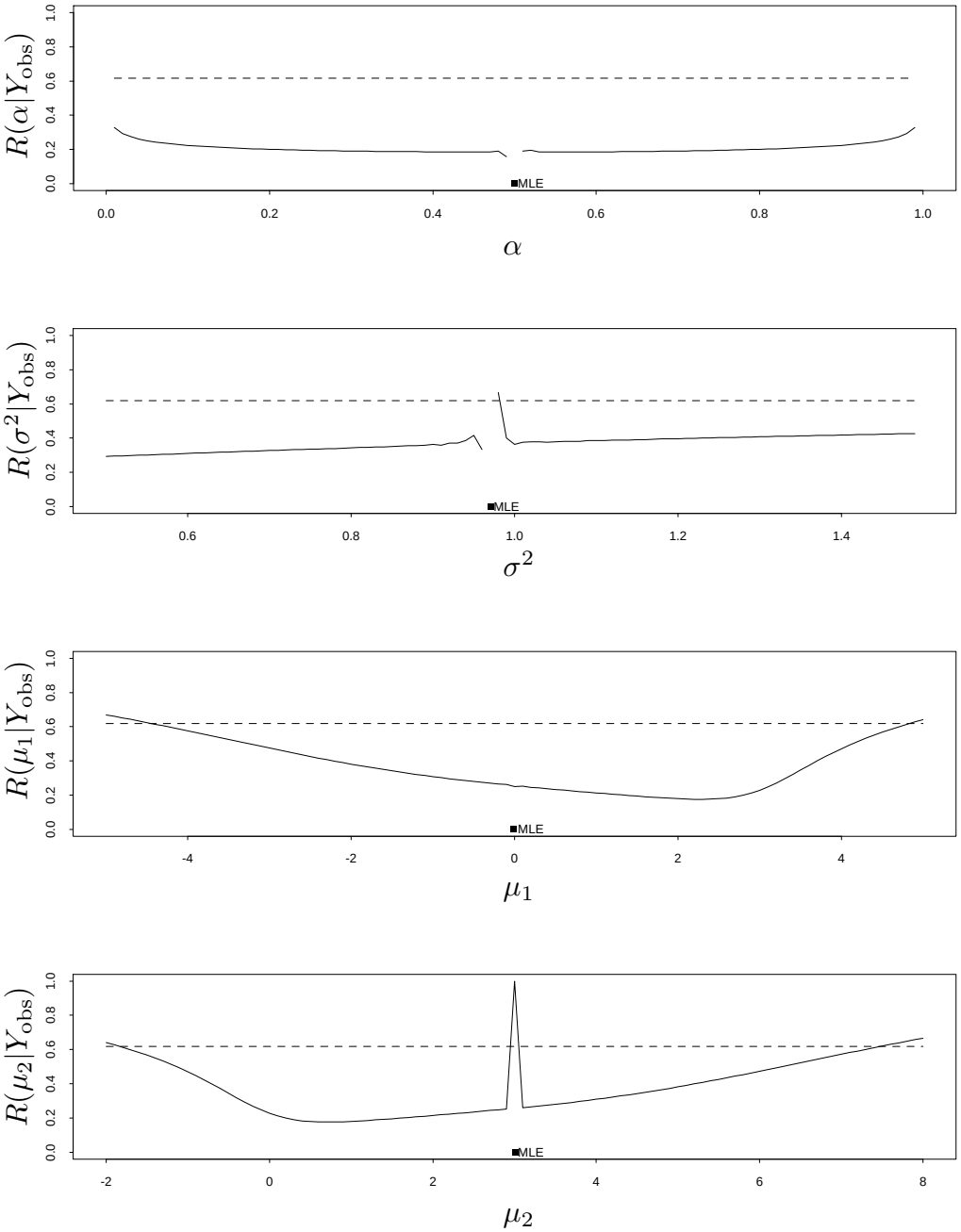

Figure 6.1. Plotting $R(\theta_k|Y_{\text{obs}})$ as a function of θ_k. The solid lines represent $R(\theta_k|Y_{\text{obs}})$ as a function of each of the components of θ_k with the other components fixed at their MLEs. These are compared with the dashed line which represents the rate of convergence of the EM algorithm. The limit as θ_k approaches $\theta_k^\star$ along each of the axes are rates of convergence of the conditional EM algorithms and are different.

(2) $\sup_{u\neq 0}\lim_{\epsilon\to 0} R(\theta^\star+\epsilon u|Y_{\rm obs}) = r$, the largest eigenvalue of $DM^{EM}(\theta^\star)$.

(3) $\lim_{\tilde\theta_i\to\tilde\theta_i^\star} R(\tilde\theta_1^\star,\ldots,\tilde\theta_{i-1}^\star,\tilde\theta_i,\tilde\theta_{i+1}^\star,\ldots,\tilde\theta_d^\star|Y_{\rm obs}) = \tilde r_i$, where $\tilde\theta_i$ is the i th component of θ and $\tilde r_i$ is the rate of convergence of the EM algorithm whose M-step only updates $\tilde\theta_i$ conditional on the other components of $\theta^\star$. These conditional rates are just the ratio of the (i,i) diagonal elements of $I_{\rm mis}$ and $I_{\rm aug}$.

(4) $r\geq\tilde r_i$ for each i .

Proof:

(1) Let $H(\theta|\theta^\star) = Q(\theta|\theta^\star) - L(\theta|Y_{\rm obs})$. It is known that $H(\theta^\star|\theta^\star)\geq H(\theta|\theta^\star)$ for any $\theta\in\Theta$, (Dempster, Laird, and Rubin, 1977), and thus

$$\begin{aligned} Q(\theta^\star|\theta^\star) - Q(\theta|\theta^\star) &= [L(\theta^\star|Y_{\rm obs}) - L(\theta|Y_{\rm obs})] + [H(\theta^\star|\theta^\star) - H(\theta|\theta^\star)] \\ &\geq L(\theta^\star|Y_{\rm obs}) - L(\theta|Y_{\rm obs}) \geq 0, \end{aligned}$$

which implies (1).

(2) Using Taylor's expansion (assuming the required derivatives exist)

$$\begin{aligned} L(\theta^\star+\epsilon u|Y_{\rm obs}) = L(\theta^\star|Y_{\rm obs}) + \\ \epsilon u^\top\left[\nabla L(\theta|Y_{\rm obs})\big|_{\theta=\theta^\star}\right] + \frac{1}{2}\epsilon u^\top\left[\nabla^2 L(\theta|Y_{\rm obs})\big|_{\theta=\theta^\star}\right]\epsilon u + o(\epsilon^3) \end{aligned}$$

and similar expansion for $Q(\theta^\star+\epsilon u|\theta^\star)$, and remembering that the gradients evaluated at $\theta^\star$ are zero we obtain

$$\begin{aligned} R(\theta^\star+\epsilon u|Y_{\rm obs}) &= 1 - \frac{u^\top\left[\nabla^2 L(\theta|Y_{\rm obs})|_{\theta=\theta^\star}\right]u + o(\epsilon)}{u^\top\left[\nabla^2 Q(\theta|\theta^\star)|_{\theta=\theta^\star}\right]u + o(\epsilon)} \\ &= 1 - \frac{u^\top I_{\rm obs}u + o(\epsilon)}{u^\top I_{\rm aug}u + o(\epsilon)}. \end{aligned} \tag{6.4.2}$$

It follows that,

$$\sup_{u\neq 0}\lim_{\epsilon\to 0} R(\theta^\star+\epsilon u|Y_{\rm obs}) = \sup_{u\neq 0}\frac{u^\top[I_{\rm aug}-I_{\rm obs}]u}{u^\top I_{\rm aug}u} = \sup_{u\neq 0}\frac{u^\top I_{\rm mis}u}{u^\top I_{\rm aug}u}, \tag{6.4.3}$$

which is the largest eigenvalue of $I_{\text{mis}} I_{\text{aug}}^{-1}$ since $I_{\text{aug}} > 0$.

(3) This is a simple extension of (2). If we let $\vartheta = (\tilde{\theta}_1, \ldots, \tilde{\theta}_{i-1}, \tilde{\theta}_{i+1}, \ldots, \tilde{\theta}_d)$ and condition on $\vartheta = \vartheta^\star$, we are left with the one parameter model with likelihood $L(\theta_i | Y_{\text{obs}}, \vartheta^\star)$. The proof goes through as before except that u is a scalar and cancels in (6.4.3) so that the sup operation is unnecessary. Thus, $\lim_{\tilde{\theta}_i \to \tilde{\theta}_i^\star} R(\tilde{\theta}_1^\star, \ldots, \tilde{\theta}_{i-1}^\star, \tilde{\theta}_i, \tilde{\theta}_{i+1}^\star, \ldots, \tilde{\theta}_d^\star | Y_{\text{obs}})$ converges to the ratio of the 1×1 conditional information matrices which is exactly the rate of convergence of the conditional EM algorithm.

(4) This is a simple consequence of (2) and (3) by using (6.4.3)

$$r = \sup_{u \neq 0} \frac{u^\top I_{\text{mis}} u}{u^\top I_{\text{aug}} u} \geq \frac{e^\top I_{\text{mis}} e}{e^\top I_{\text{aug}} e} = \tilde{r}_i, \tag{**6.4.4**}$$

where $e^\top = (0, \ldots, 0, 1, 0, \ldots, 0)$ with i^{th} element 1.

■

Property (2) makes clear what criterion $\hat{k}_r$ uses to estimate k. The EM algorithm is designed to formulate $Q(\theta_k | \theta_k^\star)$ so that the same value of θ_k will maximize it and $L(\theta_k | Y_{\text{obs}})$. The estimate $\hat{k}_r$ tries to take this one step further, by choosing the value of k for which $\nabla^2 L(\theta_k^\star | Y_{\text{obs}})$ is closest to $\nabla^2 Q(\theta_k^\star | \theta_k^\star)$ (or by minimizing (6.4.3)); namely, it attempts to match the curvature of the two likelihood surfaces. In other words, $\hat{k}_r$ can be viewed as a minimax estimator in that it minimizes the maximum directional loss of information in a neighborhood of $\theta_k^\star$.

$$\min_k \sup_{u \neq 0} \lim_{\epsilon \to 0} R(\theta_k^\star + \epsilon u | Y_{\text{obs}}) = \sup_{u \neq 0} \lim_{\epsilon \to 0} R(\theta_{\hat{k}_r}^\star + \epsilon u | Y_{\text{obs}}).$$

This minimax formulation not only helps us to understand the procedure better, but also helps to establish the underestimating property of $\hat{k}_r$ as detailed

in the next sections. Although we could have discussed this minimax in terms of $DM_n^{EM}(k|k_0)$ directly by using the first equality in (6.4.4), the formulation of $R(\theta|Y_{\text{obs}})$ is interesting in its own right, lends insight into Windham and Cutler's approach and may lead to improved estimation procedures as discussed in Section 6.6.

6.5. Large-Sample Behavior of $\hat{k}_r$ – Empirical Results

Based on Windham and Cutler's results, we suspect that $\hat{k}_r$ may tend to underestimate the actual number of subpopulations, k_0. In this section we will provide simulation evidence that $\hat{k}_r$ indeed underestimates k_0. In order to show this, our strategy is to approximate $r(k|k_0)$ by $r_n(k|k_0)$ when n is large enough so that we can ignore the error in this approximation, and then to show that $\text{argmin}_{k\geq 2}\{\hat{r}_n(k|k_0)\}$ is not necessarily k_0. To accomplish this we ran a simulation which drew n observations from $Y_{\text{obs}} \sim \frac{1}{3}\phi(0,1) + \frac{1}{3}\phi(\delta,1) + \frac{1}{3}\phi(2\delta,1)$. The simulation was repeated for 90 values of δ between 1 and 10 and for $n = 1000, 10\,000$ and $100\,000$. For each data set EM was run to fit the model

$$Y_{\text{obs}} \sim \sum_{j=1}^{k} \alpha_j \phi(\mu_j, 1), \qquad \sum \alpha_j = 1, \tag{6.5.1}$$

with $k = 2$ and $\hat{r}_n(2|3)$ was calculated and is plotted for each n as a function of δ in Figure 6.2 (the solid line corresponds to $n = 100\,000$). The plot demonstrates that $\hat{r}_n(2|3)$ is converging as n increases and that $\hat{r}_{100\,000}(2|3)$ offers a reasonable Monte Carlo approximation of $r(2|3)$ (i.e., the following comparison can be

regarded as free of finite-sample variation).

For the second step, we will show that $\hat{k}_r \neq k_0 = 3$ for some models, thus showing that the estimate is not consistent. Consider a simulation similar to the one presented in Figure 6.2 except that we fit the correct model in equation (6.5.1) with $k = 3$. Again, we can run EM and plot $\hat{r}_{100\,000}(3|3)$ as a function of δ. This function is superimposed on $\hat{r}_{100\,000}(2|3)$ in Figure 6.3. The dashed line corresponds to $k = 2$ and the solid line corresponds to $k = 3$ (the correct model). We see that for $\delta < 3.3$ (approximately) the procedure will tend to choose the incorrect value $\hat{k}_r = 2$ since $\hat{r}_{100\,000}(2|3) < \hat{r}_{100\,000}(3|3)$.

Although $\hat{k}_r$ is not always consistent, it will be consistent if $r(k|k_0)$ is minimized when $k = k_0$. Whether or not this occurs depends on the distribution of the data and the model that is being fit. Thus, it is useful to explore the behavior of $r(k|k_0)$ in several common situations. Now we will describe a set of numerical calculations of $\hat{r}_{100\,000}(k|k_0) \approx r(k|k_0)$ and attempt to characterize the estimate's behavior. The true model is taken to be

$$Y_{\text{obs}} \sim \sum_{j=1}^{k_0} \frac{1}{k_0} \phi((j-1) * \delta, 1),$$

with $k_0 = 1, \ldots, 5$ and δ varying form 0.5 to 5. We calculate $\hat{r}_{100\,000}(k|k_0)$ for the fitted model in equation (6.5.1) with $k = 2, \ldots, 6$. Figure 6.4 contains a five by five matrix of plots of $\hat{r}_{100\,000}(k|k_0)$ versus δ. The rows of Figure 6.4 correspond to the data models (values of k_0) and the columns correspond to the fitted models (values of k). It is clear that the plots on or above the diagonal correspond to very slow algorithms, regardless of the value of δ. All of these fit more subpopulations than occur in the data. When the correct value is fit, there is a similar pattern for

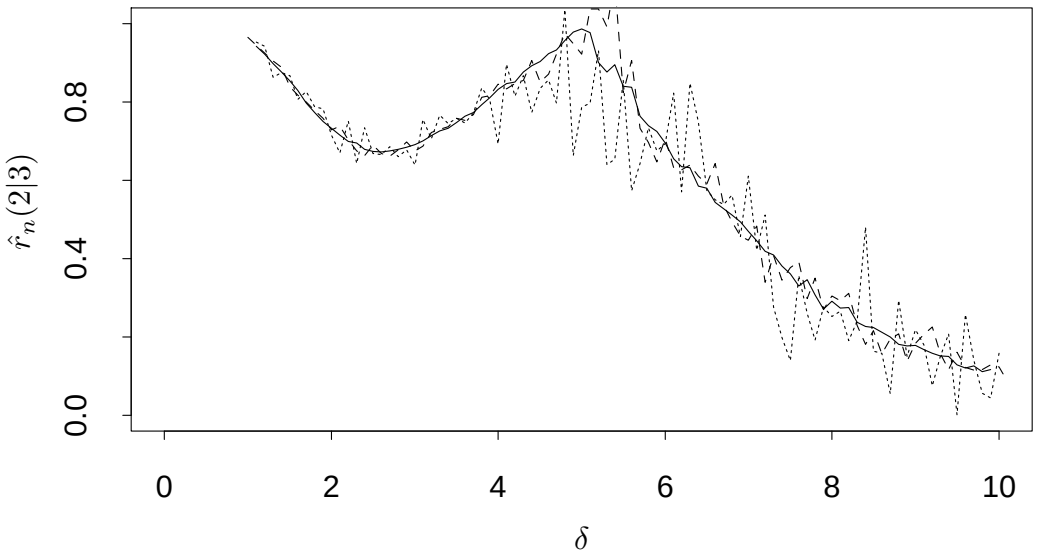

Figure 6.2. The convergence of $\hat{r}_n(k|k_0)$. The three lines show $\hat{r}_n(2|3)$ as a function of δ for several values of n (dotted: $n = 1000$, dashed: $n = 10\,000$, solid: $n = 100\,000$). The convergence of $\hat{r}_n(2|3)$ as n increases is clear.

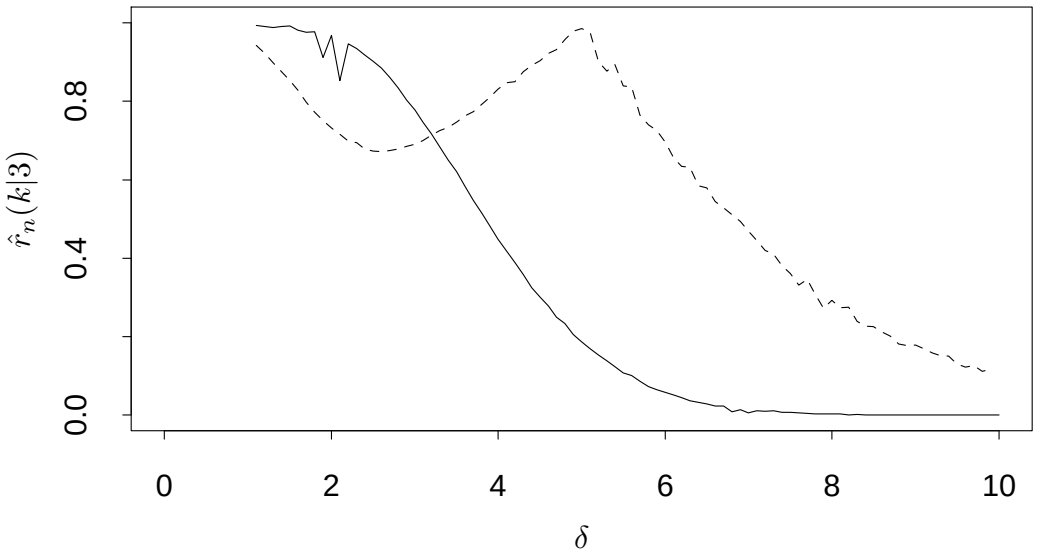

Figure 6.3. The estimator $\hat{k}_r$ is not consistent. The solid line approximates $r(3|3)$ and the dashed line $r(2|3)$. Since $r(2|3) < r(3|3)$ for $\delta < 3.3$ the estimator $\hat{k}_r$ is not consistent.

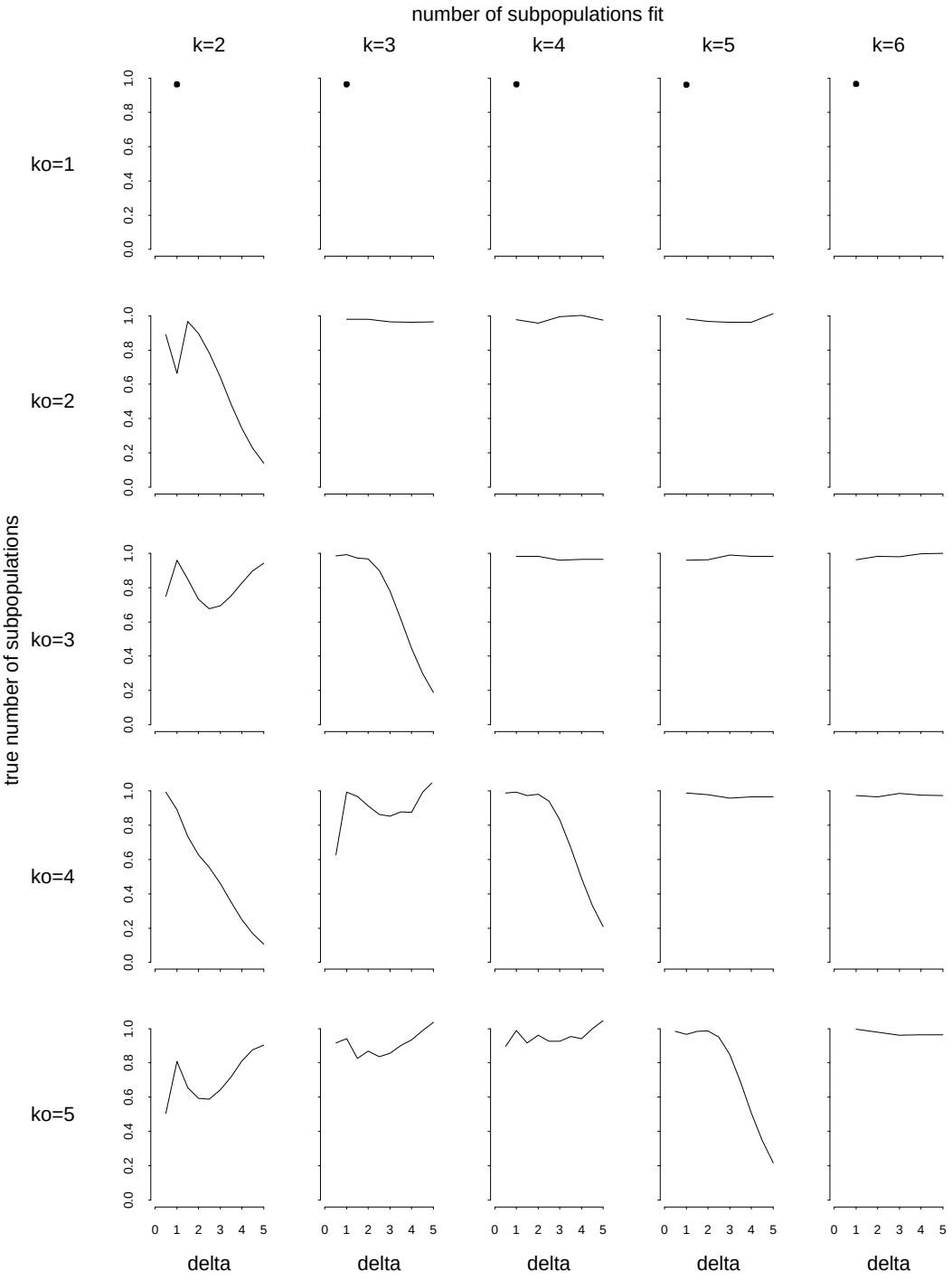

Figure 6.4. ANOVA. The plots analyze the effect of the true number of subpopulations (k_0), the fitted number of subpopulations (k), and δ on $r(k|k_0)$. Although $\hat{k}_r$ tends not to overestimate k_0 , it will often underestimate it when δ is small. The plots in the top row do not depend on δ since the model does not involve δ when there is only one subpopulation.

all of the values of k_0. The algorithm converges quite slowly if δ is less than about 2.5 and then begins to converge more quickly, converging rather fast at $\delta = 5$. The situation is more complicated when we fit models that are too small. It is not uncommon for these models to converge faster than the true model (especially for $k = 2$). Figure 6.5 superimposes the plots with the same value of k_0, the actual number of subpopulations. The solid lines correspond to fitting the correct model, the dotted lines to models that are too big and the dashed lines to models that are too small. It is evident from the plot that when δ is small, it is not at all uncommon for one of the smaller models to converge faster than the true model (this is especially clear when $k_0 = 4$). Thus, $\hat{k}_r$ will generally not pick the correct value of k_0 when there are several subpopulations and δ is small, no matter how much data is observed.

One final aspect of the simulations warrants our attention. It seems reasonable to expect the algorithms to converge more quickly as δ grows. This, however, is not always the case when $k < k_0$. This is especially clear for $k_0 = 3$ and $k = 2$ (see Figure 6.2), for which there are two critical points, at which the curve changes its monotonicity. To understand what is happening, remember that the data fall into three groups but the model is only fitting two. What is to be done with the data in the third group? One of two things: either it is combined with one of the other subpopulations, or it is split between both subpopulations. For small values of δ, the subpopulations overlap very much and it works well to split the middle group. As δ grows, however, this becomes more difficult and we observe the first critical point ($\delta \approx 2.5$). As δ continues to grows the subpopulations become more

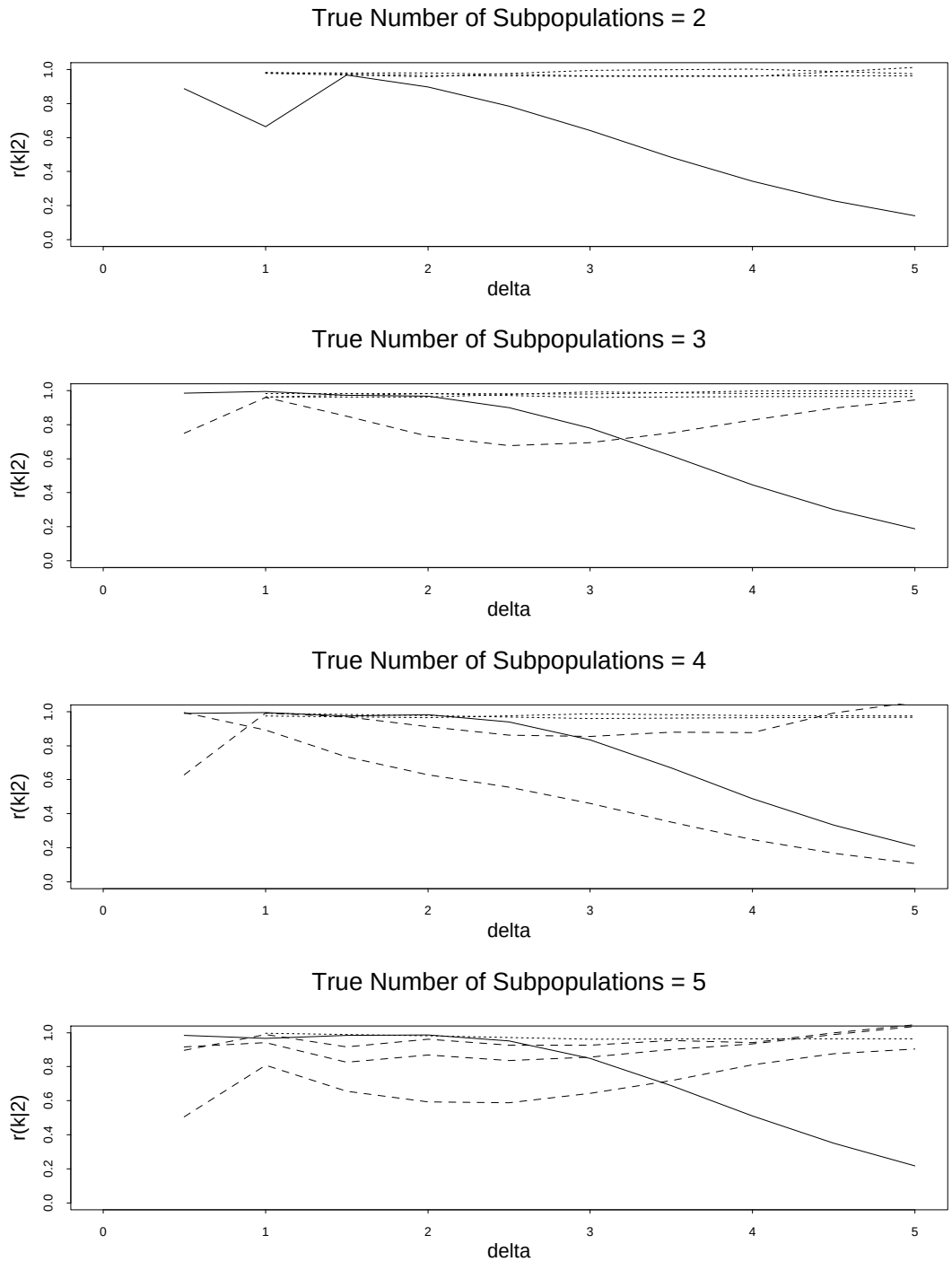

Figure 6.5. Estimating k with $\hat{k}_r$. The plots show how well different models fit for each of four values of k_0. The solid line represents the correct model, dashed lines models that are too small, and the dotted lines models that are too big. Minimizing $r(k|k_0)$ will often result in models that are too small.

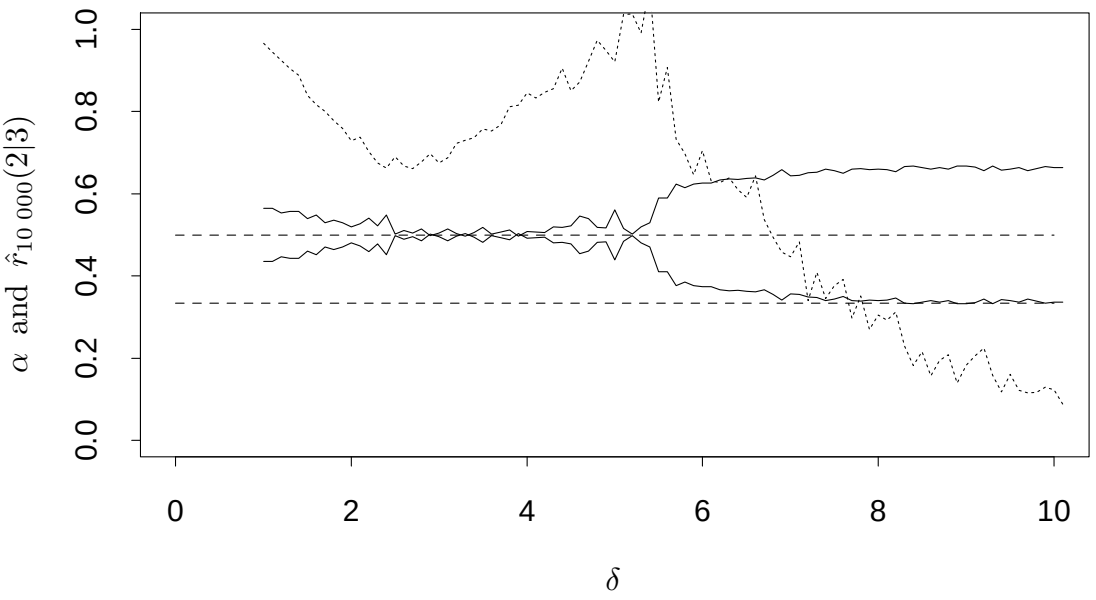

Figure 6.6. Splitting and combining subpopulations. When we fit a model that has too few components some of the subpopulations will be either split or combined. This plot shows how this is reflected in $r(k|k_0)$. The dotted line is $\hat{r}_{10\,000}(2|3)$ of Figure 6.2, the solid lines are $\alpha^\star$ and $1-\alpha^\star$, and the dashed lines are at $1/2$ and $1/3$. For $\delta < 5$ the middle subpopulation is being split, for $\delta > 5$ two subgroups are being combined.

distinct, the algorithm stops splitting the middle subpopulation and begins to combine it with one of the others, and we observe the second critical point ($\delta = 5$). This can be seen clearly by plotting the fitted value of α as a function of δ (Figure 6.6). Notice that for values of δ less than about 5, $\alpha^\star$ is near one half because the middle population is being split. For larger values of δ, $\alpha^\star$ is near one third because two subpopulations are being combined.

The empirical studies have illuminated an important characteristics of $\hat{k}_r$. It tends to underestimate the true number of subpopulations because it is influenced

not so much by k_0 as it is by the number of subpopulations salient in the observed data. This characteristic will be the topic of the next section.

6.6. Large-Sample Behavior of $\hat{k}_r$ – Discussion

In order to understand the Windham and Cutler method, we must carefully analyze the data-augmentation scheme. When the EM algorithm converges fast, the subpopulation memberships are salient in the observed data. That is, when the subpopulations have little overlap, the augmented data will contain little information that is not contained in the observed data. On the other hand when there is a lot of overlap among the subpopulations, the EM algorithm will require much data augmentation, and will converge slowly. This will be formalized by two propositions.

The propositions consider what happens to the Fisher information matrices that determine the rate of convergence of EM when the fitted number of subpopulations results in a salient separation of the data (proposition 1) and a separation that is not salient at all (proposition 2). In particular, we will consider the expected subpopulation memberships $z^\star = \{z^\star_{ij}, i = 1, \ldots, n, j = 1, \ldots, k\}$, where $z^\star_{ij} = \mathsf{E}[z_{ij}|Y_{\text{obs}}, \theta^\star]$, which can be represented by an $n \times k$ matrix with elements on the unit interval such that $\sum_{j=1}^k z^\star_{ij} = 1$ for each i. (We denote such a matrix space by $\mathcal{Z}_{n\times k}$.) By a salient separation of the data, we mean $z^\star$ is close to the boundary of $\mathcal{Z}_{n\times k}$ (i.e., matrices with only zero or one elements satisfying the constraint $\sum_j z_{ij} = 1$ for each i). A set of subpopulations which result in

a separation of the data that is far from salient can be represented by $z^\star$ "deep inside" $\mathcal{Z}_{n\times k}$, for example $z^\star_{ij} = \frac{1}{k}$ for each i and j. The two propositions look at these two extremes for $z^\star$ in detail.

Proposition 1: Suppose $f(y|\theta_k) = \sum_{j=1}^k \alpha_j g(y|\beta_j)$, where $g(y|\beta)$ is twice differentiable in the scalar parameter β, the normal equations are satisfied at the MLE of θ_k, and $z^\star$ is on the boundary of $\mathcal{Z}_{n\times k}$, then

$$I_{\text{obs}} = I_{\text{aug}}.$$

Proposition 2: Suppose $f(y|\theta_k) = \sum_{j=1}^k \alpha_j g(y|\beta_j)$, where $g(y|\beta)$ is twice differentiable in the scalar parameter β, and $z^\star$ has two rows identical, then

$$|I_{\text{obs}}| = 0.$$

The proofs of the proposition will be presented in Section 6.7. The first proposition suggests that if the subpopulations are salient in the observed data, EM will converge very quickly. The second suggests if there is no information to distinguish between two subpopulations in the observed data, EM will converge very slowly. Although both of the propositions assume the subpopulations densities differ only in the value of a scalar parameter, a close look at the proofs suggests that they could be generalized.

Several informal observations can be made based on these propositions. First, $\hat{k}_r$ does not estimate the true number of subpopulations but rather determines the number of subpopulations that results in the most salient (per the above definition) division of the data. We see this in action in Figure 6.6. Note that when δ is about

5, and the middle subpopulation is being split between the two subpopulations in the model, EM converges very slowly since the observations in the middle population could easily have come from either of the subpopulations. When δ grows a little, however, we see EM converging much more quickly as a more salient division of the data is possible – namely, combining two of the populations into one in the model. A second observation is that even with large data sets $\hat{k}_r$ will not be a good estimate of the true number of subpopulations. If two or more of the subpopulations have heavy overlap, more data will not help to separate them (i.e., the $z_{ij}^{\star}$ will not go to zero or one as n increases).

Although $\hat{k}_r$ may not be a good estimator of the number of subpopulations, it is not without merit. As we have seen, $r_n(k|k_0)$ is a good measure of how well the data can be broken into k subgroups. Trying to minimize this quantity, however, may not be the best strategy. After all, any data set falls nicely into one population, and any well separated mixture with $k_0 = 4$ will separate nicely into two or three subpopulations. Thus, a large value of k that results in reasonable convergence may be a good estimate of k_0, even if it does not result in the most salient separation of the data (i.e., minimizing $r_n(k|k_0)$). Reading across the rows of Figure 6.4 indicates that taking $\hat{k}$ as large as possible such that EM does not converge pathologically slow often results in a good estimate. To see why this is true, suppose $k > k_0$, in which case we are fitting several parameters for which there are really no observed Fisher information. Thus, $L(\theta_k|Y_{\text{obs}})$ will be very flat along these parameters, and the rate function will be nearly 1 as θ_k approaches $\theta_k^{\star}$ in these directions. Thus, by (2) of Theorem 6.1, $r_n(k|k_0)$ will be nearly 1 and EM

will be very slow to converge. For $k \leq k_0$, on the other hand, the EM algorithm generally converges reasonably well if k results in a reasonably salient separation of the data. Thus, estimating k as the largest value which does not result in pathologically slow convergence often leads to a good estimate of k_0. Of course, this procedure may not be well defined in some cases since "pathologically slow" may be a relative term. In our experience, however, fitting too many subpopulations leads to very ill-behaved values of $r_n(k|k_0)$, which not only were large but also did not converge well as t increased.

In summary, it seems likely that these procedures will tend to underestimate the true k. Although this underestimation is less than ideal, it perhaps is the best we can hope for and in fact may be a good reflection of the behavior of the observed data. As is well known, when dealing with real data, all we can hope for is to detect a few major mixture components, which fortunately is often enough for the purpose of inference. In any case, mixture models for real data can only be viewed as a useful approximation that help to identify some important underlying heterogeneous groups.

One shortcoming in both our simulations and the example in Windham and Cutler (1992) is that the variance structure was known, and not estimated from the data. There are many difficulties in implementation that are avoided when only location parameters are estimated. The likelihood for finite mixtures tends to be very badly behaved with multi-modalities and unidentifiable parameters. These difficulties tend to be more acute when scale parameters are estimated from the data. In preliminary studies in this context, we have found that there are indeed

many modes of the likelihood function and that the rate of convergence can be difficult to calculate (i.e., r_t, as defined in (1.6.9), does not converge well) and varies from mode to mode, especially when $k > k_0$. It is important, therefore, that we are not quick to generalize our findings to the unknown scale problem before more work is done.

Finally, it is important to note that the idea of using the rate of convergence to help select a model is not limited to fitting finite mixture distributions. For example, it could be used to estimate the number of factors in factor analysis. If an approach can be shown to be effective in the context of mixtures, it is likely to be useful in other applications involving estimating the number of latent variables.

6.7. Theoretical Development

6.7.1. Proof of propositions 1 and 2

Proof of Proposition 1: In this proof we will write the observed information in terms of $z_{ij}^\star$ and $\alpha_j^\star$ using the relationship

$$\frac{z_{ij}^\star}{\alpha_j^\star} = \frac{g(y_i|\beta_j^\star)}{f(y_i|\theta^\star)}, \quad j = 1, \ldots, k, \;\; i = 1, \ldots, n, \tag{6.7.1}$$

which follows from evaluating (6.2.2) at $\theta^{(t)} = \theta^\star$. We will then evaluate the observed information at $z^\star \in \mathcal{B}(\mathcal{Z}_{n\times p})$, the boundary of $\mathcal{Z}_{n\times p}$ to show that $I_{\text{aug}} = I_{\text{obs}}$. In this and the next proof, we will suppress the subscript k on the parameter θ to avoid confusion with the parameter subscripts which index the subpopulations.

We will show that $I_{\text{obs}} = I_{\text{aug}}$ term by term, beginning with:

$$\begin{aligned}
-\frac{\partial^2 L(\theta|Y_{\text{obs}})}{\partial\alpha_l\partial\alpha_m}\Big|_{\theta=\theta^\star} &\equiv I_{\text{obs}}(\alpha_l^\star,\alpha_m^\star) \qquad \text{for } l,m=1,\ldots,k-1 \\
&= \sum_{i=1}^n \frac{\big(g(y_i|\beta_l^\star)-g(y_i|\beta_k^\star)\big)\big(g(y_i|\beta_m^\star)-g(y_i|\beta_k^\star)\big)}{f^2(y_i|\theta^\star)} \\
&= \sum_{i=1}^n \left(\frac{z_{il}^\star}{\alpha_l^\star}-\frac{z_{ik}^\star}{\alpha_k^\star}\right)\left(\frac{z_{im}^\star}{\alpha_m^\star}-\frac{z_{ik}^\star}{\alpha_k^\star}\right). \qquad \textbf{(6.7.2)}
\end{aligned}$$

Evaluating (6.7.2) at $z^\star \in \mathcal{B}(\mathcal{Z}_{n\times p})$, in which case $z_{il}^\star z_{im}^\star = 0$ for $l \neq m$ (which follows form $\sum_j z_{ij}^\star = 1$), and thus

$$\begin{aligned}
I_{\text{obs}}(\alpha_l^\star,\alpha_m^\star) &= \frac{z_{il}^\star}{\alpha_l^\star}\frac{z_{im}^\star}{\alpha_m^\star} + \left(\frac{z_{ik}^\star}{\alpha_k^\star}\right)^2 \\
&= \begin{cases} \displaystyle\sum_{i=1}^n \frac{z_{ik}^\star}{\alpha_k^{\star 2}} & \text{if } l \neq m \\ \displaystyle\sum_{i=1}^n \left(\frac{z_{il}^\star}{\alpha_l^{\star 2}}+\frac{z_{ik}^\star}{\alpha_k^{\star 2}}\right) & \text{if } l = m \end{cases} \qquad \textbf{(6.7.3)}
\end{aligned}$$

which is exactly the same as the corresponding element of I_{aug}, $-\frac{\partial^2 Q(\theta|\theta^\star)}{\partial\alpha_l\partial\alpha_m}\Big|_{\theta=\theta^\star}$.

Now we will consider the terms off the block diagonal:

$$\begin{aligned}
-\frac{\partial^2 L(\theta|Y_{\text{obs}})}{\partial\alpha_l\partial\beta_m}\Big|_{\theta=\theta^\star} &\equiv I_{\text{obs}}(\alpha_l^\star,\beta_m^\star) \qquad \text{for } l=1,\ldots,k-1 \;\; m=1,\ldots,k \\
&= \alpha_m^\star \sum_{i=1}^n \left(\frac{g(y_i|\beta_l^\star)-g(y_i|\beta_k^\star)}{f^2(y_i|\theta^\star)}\right)\frac{\partial g(y_i|\beta_m^\star)}{\partial\beta_m} \\
&= \alpha_m^\star \sum_{i=1}^n \left(\frac{z_{il}^\star}{\alpha_l^\star}-\frac{z_{ik}^\star}{\alpha_k^\star}\right)\frac{z_{im}^\star}{\alpha_m^\star}\frac{\partial \log g(y_i|\beta_m^\star)}{\partial\beta_m}. \qquad \textbf{(6.7.4)}
\end{aligned}$$

Evaluating (6.7.4) at $z^\star \in \mathcal{B}(\mathcal{Z}_{n\times p})$ yields

$$I_{\text{obs}}(\alpha_l^\star,\beta_m^\star) = \begin{cases} 0 & \text{if } m \neq l \text{ and } m \neq k, \\ \displaystyle\sum_{i=1}^n \frac{z_{im}^\star}{\alpha_m^\star}\frac{\partial \log g(y_i|\beta_m^\star)}{\partial\beta_m} & \text{if } m = l, \\ \displaystyle -\sum_{i=1}^n \frac{z_{im}^\star}{\alpha_m^\star}\frac{\partial \log g(y_i|\beta_m^\star)}{\partial\beta_m} & \text{if } m = k. \end{cases} \qquad \textbf{(6.7.5)}$$

But $I_{\rm obs}(\alpha_l^\star, \beta_m^\star) = 0$ when $m = l$ or $m = k$ as well since (6.7.5) is the observed data normal equation for β_m. Thus, again $I_{\rm obs}(\alpha_l^\star, \beta_m^\star) = 0$ agrees with the the corresponding $I_{\rm aug}(\alpha_l^\star, \beta_m^\star)$, which is zero.

Next we turn to β and first consider $l \neq m$,

$$\begin{aligned} -\frac{\partial^2 L(\theta|Y_{\rm obs})}{\partial\beta_l \partial\beta_m}\Big|_{\theta=\theta^\star} &\equiv I_{\rm obs}(\beta_l^\star, \beta_m^\star) \qquad \text{for } l, m = 1, \ldots, k \\ &= \sum_{i=1}^n \frac{\alpha_l^\star \alpha_m^\star \frac{\partial}{\partial\beta_l} g(y_i|\beta_l) \frac{\partial}{\partial\beta_m} g(y_i|\beta_m)}{f^2(y_i|\theta^\star)} \\ &= \sum_{i=1}^n \alpha_l^\star \alpha_m^\star z_{il}^\star z_{im}^\star \frac{\partial \log g(y_i|\beta_l^\star)}{\partial\beta_l} \frac{\partial \log g(y_i|\beta_m^\star)}{\partial\beta_m}, \end{aligned}$$

which is zero at $z^\star \in \mathcal{B}(\mathcal{Z}_{n\times p})$ since $l \neq m$, again agreeing with $I_{\rm aug}(\beta_l^\star, \beta_m^\star)$. Finally, we consider $l = m$,

$$\begin{aligned} -\frac{\partial^2 L(\theta|Y_{\rm obs})}{\partial\beta_l \partial\beta_l}\Big|_{\theta=\theta^\star} &\equiv I_{\rm obs}(\beta_l^\star, \beta_l^\star) \\ &= \sum_{i=1}^n \left[\alpha_l^{\star 2} \left(\frac{\frac{\partial}{\partial\beta_l} g(y_i|\beta_l)}{f(y_i|\theta^\star)} \right)^2 - \alpha_l^\star \frac{\frac{\partial^2}{\partial\beta_l^2} g(y_i|\beta_l^\star)}{f(y_i|\theta^\star)} \right] \\ &= \sum_{i=1}^n \left[z_{il}^{\star 2} \left(\frac{\partial \log g(y_i|\beta_l^\star)}{\partial\beta_l} \right)^2 - z_{il}^\star \frac{\frac{\partial^2}{\partial\beta_l^2} g(y_i|\beta_l^\star)}{g(y_i|\theta_l^\star)} \right], \end{aligned}$$

Once again, after noting that $z_{ij}^\star = z_{ij}^{\star 2}$, when $z^\star \in \mathcal{B}(\mathcal{Z}_{n\times p})$ the expression agrees with $I_{\rm aug}(\beta_l^\star, \beta_m^\star)$, which completes the proof.

■

Proof of Proposition 2: Without loss of generality, we assume the last row of $z^\star$ is identical to the j th row of $z^\star$, where $j \neq k$. Clearly $\alpha_j^\star = \alpha_k^\star$, since

$\alpha_j^\star = \sum_i z_{ij}^\star / n$. This fact along with (6.2.2) gives us

$$\frac{z_{ij}^\star}{\alpha_j^\star} = \frac{z_{ik}^\star}{\alpha_k^\star} \quad \text{for} \quad i = 1, \ldots, n.$$

Thus, $I_{\text{obs}}(\alpha_j^\star, \xi^\star) = 0$, where ξ is any component of θ (see (6.7.2) and (6.7.4)).

■

6.7.2. Large-sample behavior of $\hat{k}_r$ – preliminary theoretical results

Figure 6.2 gives evidence that $r_n(k|k_0) \to r(k|k_0)$. In this section we use this proposition along with a second proposition to show that with probability one the limit supremum of $\hat{k}_r$ will be no larger than k_0 , thus confirming the non-overestimation that we observed in the previous sections. After we present Theorem 6.2, we will discuss how the two new propositions, (P3) and (P4), might be proven.

Theorem 6.2: Suppose

(P3) $\lim_{n\to\infty} r_n(k|k_0) = r(k|k_0)$, almost surely for any $k \geq k_0$

(P4) If $k > k_0$ then $r(k|k_0) > r(k_0|k_0)$.

Then if $k_0 \geq 2$,

$$\limsup_{n\to\infty} \hat{k}_n \leq k_0, \quad a.s.$$

Proof: For any $k > k_0$, from (P3) and (P4) we have,

$$\mathsf{P}\left[\bigcap_{m=1}^{\infty} \bigcup_{n=m}^{\infty} \{r_n(k|k_0) \leq r_n(k_0|k_0)\}\right] = 0. \qquad (\mathbf{6.7.6})$$

Since k takes on only countably many values, (6.7.6) implies

$$\mathsf{P}\left[\bigcup_{k=k_0+1}^{\infty} \bigcap_{m=1}^{\infty} \bigcup_{n=m}^{\infty} \{r_n(k|k_0) \leq r_n(k_0|k_0)\}\right] = 0. \qquad (\mathbf{6.7.7})$$

By the definition of $\hat{k}_n$, we have

$$\{\hat{k}_n = k\} \subset \{r_n(k|k_0) \leq r_n(k_0|k_0)\},$$

and thus (6.7.7) implies

$$\mathsf{P}\left[\bigcup_{k=k_0+1}^{\infty} \bigcap_{m=1}^{\infty} \bigcup_{n=m}^{\infty} \{\hat{k}_n = k\}\right] = 0.$$

■

Now it remains to discuss propositions 3 and 4, the first of which is more of a regularity condition.

Proposition 3: If $Y_{\text{obs}} = (y_1, y_2, \ldots, y_n)$ consists of independent observations from (6.1.1) with k_0 components, then

$$\lim_{n\to\infty} r_n(k|k_0) \equiv r(k|k_0) \quad \textit{exists almost surely.}$$

In order to prove Proposition 3, by (6.4.3) of Theorem 6.3, it suffices to show that with probability 1,

$$\lim_{n\to\infty} \sup_{u\neq 0} \lim_{\epsilon\to 0} R(\theta_k^\star + \epsilon u|Y_{\text{obs}}) = \lim_{n\to\infty} \sup_{u\neq 0} \left\{\frac{u^\top I_{\text{mis}}(n)u}{u^\top I_{\text{aug}}(n)u}\right\} = \sup_{u\neq 0} \lim_{n\to\infty} \left\{\frac{u^\top I_{\text{mis}}(n)u}{u^\top I_{\text{aug}}(n)u}\right\}$$

exists. If $\theta_k^\star$ is consistent, we should be able to prove this without too much difficulty. The technical difficulty here involves dealing with the possibility that I_{aug} may be ill-behaved when $\theta_k^\star$ converges to a boundary value of Θ_k. This problem will be the subject of future work. The next proposition will likely prove to be even more challenging.

Proposition 4: If $k > k_0$, then $r(k|k_0) > r(k_0|k_0)$.

This proposition is at the heart of Theorem 6.2 and its proof has many subtleties. For the moment we will prove a related result that we hope will ultimately lead to Proposition 4.

Proposition 4′: Let $k > k_0$ and $\theta_k = (\theta_{k_0}, \xi)$, where ξ are the additional parameters the model $f(y|\theta_k)$ has compared with $f(y|\theta_{k_0})$. Then

$$r_n(k|k_0) \geq r'_n(k_0|k_0, \xi = \xi^\star),$$

where $r'_n(k_0|k_0, \xi = \xi^\star)$ is the rate of convergence of the EM algorithm that calculates $\theta^\star_{k_0}$ conditional on $\xi = \xi^\star$, where $\theta^\star_k = (\theta^\star_{k_0}, \xi^\star)$ is the MLE of θ_k from fitting $f(y|\theta_k)$.

Proof: This result can be seen from (6.4.3):

$$r_n(k|k_0) = \sup_{u \neq 0} \left\{ \frac{u^\top I_{\text{mis}} u}{u^\top I_{\text{aug}} u} \right\} \geq \sup_{u \neq 0, u \in \mathcal{U}} \left\{ \frac{u^\top I_{\text{mis}} u}{u^\top I_{\text{aug}} u} \right\} = r'_n(k_0|k_0, \xi = \xi^\star),$$

where $\mathcal{U}$ is a subspace of $\mathbb{R}^d$ with all elements corresponding to ξ equal to zero.

■

The relevance of Proposition 4′ to Proposition 4 lies in showing

$$\lim_{n \to \infty} r'_n(k_0|k_0, \xi = \xi^\star) = r'(k_0|k_0, \xi = 0),$$

the final term of which is $r(k_0|k_0)$ ($\lim_{n\to\infty} r_n(k|k_0) = r(k|k_0)$ is guaranteed by Proposition 3). The reason that we can expect this to be true is as follows. If we fit a model with $k \geq k_0$, the true model is contained in the subspace Θ_0 of Θ_k, where θ_k has $\alpha_j = 0$ for $j \geq k_0 + 1$. If maximum likelihood estimates are consistent, then $\theta^\star_k$ will converge to a value in Θ_0. Thus, the conditional EM

algorithm with model $f(Y_{\text{obs}}|\theta_k)$ should converge at the same rate, asymptotically as the unconditional EM algorithm with model $f(Y_{\text{obs}}|\theta_{k_0})$. We must be cautious, however, since Θ_0 is contained in the boundary of Θ_k and thus the standard consistency arguments need to be carefully checked. Even more important, (6.4.3) assumes that $\nabla L(\theta_k|Y_{\text{obs}}) = 0$ at $\theta_k = \theta_k^\star$. However, the fact that $\theta_k^\star$ converges to a boundary point of Θ_k implies that this condition might not hold, a possibility that will complicate the theoretical derivation here.

References

Aitkin, M., Anderson, D. and Hinde, J. (1981). Statistical modeling of data on teaching styles. *J. R. Statist. Soc.* **A, 144,** 419-461.

Aitkin, M., and Rubin, D. B. (1985). Estimation and hypothesis testing in finite mixture models. *J. R. Statist. Soc.* **B, 47** 67-75.

Amemiya, T. (1984). Tobit models: a survey. *J. Econometrics* **24**, 3-61.

Amit, Y. and Grenander, U. (1991). Comparing sweep strategies for stochastic relaxation. *J. of Multivariate Analysis.* **37,** 197-222.

Beaton, A. E. (1964). The use of special matrix operations in statistical calculus. Education Testing Service Research Bulletin, RB-64-51.

Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice.* MIT Press, Cambridge, Massachusetts.

Carnahan, B., Luther, H. A. and Wilkes, J. O. (1969). *Applied Numerical Methods.* John Wiley & Sons, New York.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood estimation from incomplete-data via the EM algorithm (with discussion). *J. R. Statist. Soc.* **B, 39,** 1-38.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1980). Iteratively reweighted least squares for linear regression when errors are normal/independent distributed. *Multivariate Analysis* **V,** 35-37.

Dyk, D. A. van (1993). Fitting log-linear models to contingency tables with incomplete data. *Technical Report 381*, Department of Statistics, University of Chicago.

Dyk, D. A. van and Meng, X. L. (1994). Permuting CM steps within the ECM algorithm: implementational strategies and cautions. *Technical Report 397*, Department of Statistics, University of Chicago. Submitted to *J. Computat. Graph. Statist.*

Dyk, D. A. van, Meng, X. L. and Rubin, D. B. (1994). Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance. *Technical Report 380.* Dept. of Statistics, University of Chicago.

Dyk, D. A. van, Meng, X. L. and Rubin, D. B. (1995). Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance. *Statistica Sinica* **5,** 55-75.

Fessler, J. A. and Hero, A. O. (1994). Space-alternating generalized expectation-maximization algorithm. *IEEE Tran. on Signal Processing* **42** 2664-77.

Henna, J. (1985). On estimation of countable mixtures of continuous distributions. *J. of the Japan. Statist. Soc.* **15,** 75-82.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis.* Cambridge University Press, New York.

Jamshidian, M. and Jennrich, R. I. (1993). Conjugate gradient acceleration of the EM algorithm. *J. Am. Statist. Assoc.* **88,** 221-228.

Kayalar, S. and Weinert, H. L. (1988). Error bounds for the method of alternating projections. *Math. Control Signals Systems* **1,** 43-59.

Kent, J. T. and Tyler, D. E. (1991). Redescending M-estimates of multivariate location and scatter. *Ann. Statist.* **19,** 2102-2119.

Kent, J. T., Tyler, D. E. and Vardi, Y. (1994). A curious likelihood identity for the multivariate t-distribution. *Commun. Statist. – Simula.* **23,** 441-453.

Laird, N., Lange, N. and Stram, D. (1987). Maximizing likelihood computations with repeated measures: application of the EM algorithm. *J. Am. Statist. Assoc.* **82,** 97-105.

Laird, N. M. and Ware, J. H. (1982). Random effects models for longitudinal data. *Biometrics* **38,** 967-74.

Lange, K. (1995). A quasi-Newtonian acceleration of the EM algorithm. *Statistica Sinica* **5,** 1-18.

Lange K., Little, R. J. A. and Taylor, J. M. G. (1989). Robust statistical modeling using the t-distribution. *J. Am. Statist. Assoc.* **84,** 881-896.

Lansyk, D. and Casella, G. (1990). Improving the EM algorithm. *Computing Science and Statistics: Proceedings of the Symposium on the Interface*, 420-424.

Lindstrom, M. J. and Bates, D. M. (1988). Newton-Raphson and EM algorithms for linear mixed-effects models for repeated-measure data. *J. Am. Statist. Assoc.* **83,** 1014-1022.

Little, R. J. A. (1988). Robust estimation of the mean and covariance matrix from data with missing values. *Applied Statistics*, **37,** 23-39.

Little, R. J. A. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. John Wiley & Sons, New York.

Liu, C. and Rubin, D. B. (1995a). The ECME algorithm: a simple extension of EM and ECM with fast monotone convergence. *Biometrika*, **81,** 633-48.

Liu, C. and Rubin, D. B. (1995b). ML estimation of the t-distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, **5,** 19-40.

Louis, T. A., (1982). Finding the observed information matrix when using the EM algorithm. *J. R. Statist. Soc.* **B, 44,** 226-233.

Maine, M., Boullion, T. and Rizzuto, G. T. (1991). Detecting the number of components in a finite mixture having normal components. *Commun. Statist. –Theory Meth.* **20,** 611-620.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). *Multivariate Analysis.* Academic Press, London.

Meng, X. L. (1990). Towards complete results for some incomplete-data Problems. Ph.D. Thesis, Harvard University, Department of Statistics.

Meng, X. L. (1994). On the rate of convergence of the ECM algorithm. *Ann. Statist.* **22,** 326-339.

Meng, X. L. and Pedlow, S. (1992). EM: A bibliographic review with missing articles. *Proc. Statist. Comp. Sect.,* 24-27. Washington, D.C.: American Statistical Association.

Meng, X. L. and Rubin, D. B. (1991a). Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. *J. Am. Statist. Assoc.* **86,** 899-909.

Meng, X. L. and Rubin, D. B. (1991b). IPF for contingency tables with missing data via the ECM algorithm. *Proc. Statist. Comp. Sect.,* 244-247. Washington, D.C.: American Statistical Association.

Meng, X. L. and Rubin, D. B. (1992). Recent extensions to the EM algorithm (with discussion). In *Bayesian Statistics 4,* Ed. J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, pp. 307-20. Oxford University Press.

Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika* **80,** 267-78.

Meng, X. L. and Rubin, D. B. (1994a). On the global and componentwise rates of convergence of the EM algorithm. *Linear Algebra and its Applications (Special issue honoring Ingram Olkin)* **199,** 413-425.

Meng, X. L. and Rubin, D. B. (1994b). Efficient methods for estimating and testing with seemingly unrelated regressions in the presence of latent variables and missing data. *To appear in a special volume in honor of Arnold Zellner.*

Orchard, T. and Woodbury, M. A. (1972). A missing information principle theory and application. *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability* **1,** 697-715.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative Solutions of Nonlinear Equations in Several Variables.* Academic Press, New York.

Pearson, K. (1894). Contribution to the mathematical theory of evolution. *Phil. Trans. Roy. Soc.* **A, 185,** 71-110.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63,** 581-592.

Rubin, D. B. (1983). Iteratively reweighted least squares, *Encyclopedia of the Statistical Sciences, Vol. 4.* John Wiley & Sons, New York, pp. 272-275.

Segal, M. R., Bacchetti, P., and Jewell, N. P. (1994). Variances for maximum penalized likelihood estimates obtained via the EM algorithm. *J. R. Statist. Soc.* **B, 56,** 345-352.

Titterington, D. M., Smith A. F. M. and Markov U. E. (1985). *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, New York.

Wei, C. G. and Tanner, M. A. (1990). A monte carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Am. Statist. Assoc.* **85,** 699-704.

Windham, M. P. and Cutler, A. (1992). Information ratios for validating mixture analyses. *J. Am. Statist. Assoc.* **87,** 1188-1192.

Wolfe, J. H. (1970). Pattern clustering by multivariate mixture analysis. *Multivar. Behav. Res.,* **5,** 329-350.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *Ann. Statist.* **11,** 95-103.

Zangwill, W. (1969). *Nonlinear Programming – A Unified Approach.* Prentice-Hall, Englewood Cliffs, New Jersey.

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *J. Am. Statist. Assoc.* **57,** 348-368.