Chapter 6
The EM Algorithm

Shu Kay Ng, Thriyambakam Krishnan, and Geoffrey J. McLachlan

6.1 Introduction

The Expectation-Maximization (EM) algorithm is a broadly applicable approach to the iterative computation of maximum likelihood (ML) estimates, useful in a variety of incomplete-data problems. It is based on the idea of solving a succession of simpler problems that are obtained by augmenting the original observed variables (the incomplete data) with a set of additional variables that are unobservable or unavailable to the user. These additional data are referred to as the missing data in the EM framework. The EM algorithm is closely related to the *ad hoc* approach to estimation with missing data, where the parameters are estimated after filling in initial values for the missing data. The latter are then updated by their predicted values using these initial parameter estimates. The parameters are then re-estimated, and so on, proceeding iteratively until convergence. On each iteration of the EM algorithm, there are two steps called the Expectation step (or the E-step) and the Maximization step (or the M-step). The name "EM algorithm" was given by Dempster et al. (1977) in their fundamental paper.

The EM algorithm has a number of desirable properties, such as its numerical stability, reliable global convergence, and simplicity of implementation. However, the EM algorithm is not without its limitations. In its basic form, the EM algorithm

S.K. Ng (✉)
School of Medicine, Griffith University, Meadowbrook, QLD 4131, Australia
e-mail: s.ng@griffith.edu.au

T. Krishnan
Mu-Sigma Business Solutions Pvt. Ltd, Kalyani Platina, K.R. Puram Hobli, Bangalore, India
e-mail: krishnant001@gmail.com

G.J. McLachlan
Department of Mathematics, University of Queensland, Brisbane, QLD, Australia
e-mail: g.mclachlan@uq.edu.au

J.E. Gentle et al. (eds.), *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-21551-3_6,

lacks of an in-built procedure to compute the covariance matrix of the parameter estimates and it is sometimes very slow to converge. Moreover, certain complex incomplete-data problems lead to intractable E-steps and M-steps. The first edition of the book chapter published in 2004 covered the basic theoretical framework of the EM algorithm and discussed further extensions of the EM algorithm to handle complex problems. The second edition attempts to capture advanced developments in EM methodology in recent years. In particular, there are many connections between the EM algorithm and Markov chain Monte Carlo algorithms. Furthermore, the key idea of the EM algorithm where a function of the log likelihood is maximized in a iterative procedure occurs in other optimization procedures as well, leading to a more general way of treating EM algorithm as an optimization procedure. Capturing the above developments in the second edition has led to the addition of new examples in the applications of the EM algorithm or its variants to complex problems, especially in the related fields of biomedical and health sciences.

The remaining of Sect. 6.1 focusses on a brief description of ML estimation and the incomplete-data structure of the EM algorithm. The basic theoretical framework of the EM algorithm is presented in Sect. 6.2. In particular, the monotonicity of the algorithm, convergence, and rate of convergence properties are systematically examined. In Sect. 6.3, the EM methodology presented in this chapter is illustrated in some commonly occurring situations such as the fitting of normal mixtures and missing observations in terms of censored failure times. Another example is provided in which the EM algorithm is used to train a mixture-of-experts model. Consideration is given also to clarify some misconceptions about the implementation of the E-step, and the important issue associated with the use of the EM algorithm, namely the provision of standard errors. We discuss further modifications and extensions to the EM algorithm in Sect. 6.4. In particular, the extensions of the EM algorithm known as the Monte Carlo EM, ECM, ECME, AECM, and PX–EM algorithms are considered. With the considerable attention being given to the analysis of large data sets, as in typical data mining applications, recent work on speeding up the implementation of the EM algorithm is discussed. These include the IEM, SPIEM, and the use of multiresolution kd-trees. In Sect. 6.5, the relationship of the EM algorithm to other data augmentation techniques, such as the Gibbs sampler and MCMC methods is presented briefly. The Bayesian perspective is also included by showing how the EM algorithm and its variants can be adapted to compute the maximum *a posteriori* (MAP) estimate. We conclude the chapter with a brief account of the applications of the EM algorithm in such topical and interesting areas as bioinformatics and health sciences.

6.1.1 Maximum Likelihood Estimation

Maximum likelihood estimation and likelihood-based inference are of central importance in statistical theory and data analysis. Maximum likelihood estimation is a general-purpose method with attractive properties. It is the most-often used

estimation technique in the frequentist framework, and it can be equally applied to find the mode of the posterior distribution in a Bayesian framework (Chap. III.26). Often Bayesian solutions are justified with the help of likelihoods and maximum likelihood estimates (MLE), and Bayesian solutions are similar to penalized likelihood estimates. Maximum likelihood estimation is an ubiquitous technique and is used extensively in every area where statistical techniques are used.

We assume that the observed data $\boldsymbol{y}$ has probability density function (p.d.f.) $g(\boldsymbol{y}; \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is the vector containing the unknown parameters in the postulated form for the p.d.f. of $\boldsymbol{Y}$. Our objective is to maximize the likelihood $L(\boldsymbol{\Psi}) = g(\boldsymbol{y}; \boldsymbol{\Psi})$ as a function of $\boldsymbol{\Psi}$, over the parameter space $\boldsymbol{\Omega}$. That is,

$$\partial L(\boldsymbol{\Psi})/\partial \boldsymbol{\Psi} = \mathbf{0},$$

or equivalently, on the log likelihood,

$$\partial \log L(\boldsymbol{\Psi})/\partial \boldsymbol{\Psi} = \mathbf{0}. \tag{6.1}$$

The aim of ML estimation is to determine an estimate $\hat{\boldsymbol{\Psi}}$, so that it defines a sequence of roots of (6.1) that is consistent and asymptotically efficient. Such a sequence is known to exist under suitable regularity conditions (Cramér 1946). With probability tending to one, these roots correspond to local maxima in the interior of $\boldsymbol{\Omega}$. For estimation models in general, the likelihood usually has a global maximum in the interior of $\boldsymbol{\Omega}$. Then typically a sequence of roots of (6.1) with the desired asymptotic properties is provided by taking $\hat{\boldsymbol{\Psi}}$ to be the root that globally maximizes $L(\boldsymbol{\Psi})$; in this case, $\hat{\boldsymbol{\Psi}}$ is the MLE. We shall henceforth refer to $\hat{\boldsymbol{\Psi}}$ as the MLE, even in situations where it may not globally maximize the likelihood. Indeed, in some of the examples on mixture models (McLachlan and Peel 2000, Chap. 3), the likelihood is unbounded. However, for these models there may still exist under the usual regularity conditions a sequence of roots of (6.1) with the properties of consistency, efficiency, and asymptotic normality (McLachlan and Basford 1988, Chap. 12).

When the likelihood or log likelihood is quadratic in the parameters as in the case of independent normally distributed observations, its maximum can be obtained by solving a system of linear equations in parameters. However, often in practice the likelihood function is not quadratic giving rise to nonlinearity problems in ML estimation. Examples of such situations are: (a) models leading to means which are nonlinear in parameters; (b) despite a possible linear structure, the likelihood is not quadratic in parameters due to, for instance, non-normal errors, missing data, or dependence.

Traditionally ML estimation in these situations has been carried out using numerical iterative methods of solution of equations such as the Newton–Raphson (NR) method and its variants like Fisher's method of scoring. Under reasonable assumptions on $L(\boldsymbol{\Psi})$ and a sufficiently accurate starting value, the sequence of iterates $\{\boldsymbol{\Psi}^{(k)}\}$ produced by the NR method enjoys local quadratic convergence to a solution $\boldsymbol{\Psi}^*$ of (6.1). Quadratic convergence is regarded as the major strength of

the NR method. But in applications, these methods could be tedious analytically and computationally even in fairly simple cases; see McLachlan and Krishnan (2008, Sect. 1.3) and Meng and van Dyk (1997). The EM algorithm offers an attractive alternative in a variety of settings. It is now a popular tool for iterative ML estimation in a variety of problems involving missing data or incomplete information.

6.1.2 Idea Behind the EM Algorithm: Incomplete-Data Structure

In the application of statistical methods, one is often faced with the problem of estimation of parameters when the likelihood function is complicated in structure resulting in difficult-to-compute maximization problems. This difficulty could be analytical or computational or both. Some examples are grouped, censored or truncated data, multivariate data with some missing observations, multiway frequency data with a complex cell probability structure, and data from mixtures of distributions. In many of these problems, it is often possible to formulate an associated statistical problem with the same parameters with "augmented data" from which it is possible to work out the MLE in an analytically and computationally simpler manner. The augmented data could be called the "complete data" and the available data could be called the "incomplete data", and the corresponding likelihoods, the "complete-data likelihood" and the "incomplete-data likelihood", respectively. The EM Algorithm is a generic method for computing the MLE of an incomplete-data problem by formulating an associated complete-data problem, and exploiting the simplicity of the MLE of the latter to compute the MLE of the former. The augmented part of the data could also be called "missing data", with respect to the actual incomplete-data problem on hand. The missing data need not necessarily be missing in the practical sense of the word. It may just be a conceptually convenient technical device. Thus the phrase "incomplete data" is used quite broadly to represent a variety of statistical data models, including mixtures, convolutions, random effects, grouping, censoring, truncated and missing observations.

A brief history of the EM algorithm can be found in McLachlan and Krishnan (2008, Sect. 1.8). In their fundamental paper, Dempster et al. (1977) synthesized earlier formulations of this algorithm in many particular cases and presented a general formulation of this method of finding MLE in a variety of problems. Since then the EM algorithm has been applied in a staggering variety of general statistical problems such as resolution of mixtures, multiway contingency tables, variance components estimation, factor analysis, as well as in specialized applications in such areas as genetics, medical imaging, and neural networks.

6.2 Basic Theoretical Framework of the EM Algorithm

6.2.1 The E- and M-Steps

Within the incomplete-data framework of the EM algorithm, we let $\boldsymbol{x}$ denote the vector containing the complete data and we let $\boldsymbol{z}$ denote the vector containing the missing data. Even when a problem does not at first appear to be an incomplete-data one, computation of the MLE is often greatly facilitated by artificially formulating it to be as such. This is because the EM algorithm exploits the reduced complexity of ML estimation given the complete data. For many statistical problems the complete-data likelihood has a nice form.

We let $g_c(\boldsymbol{x}; \boldsymbol{\Psi})$ denote the p.d.f. of the random vector $\boldsymbol{X}$ corresponding to the complete-data vector $\boldsymbol{x}$. Then the complete-data log likelihood function that could be formed for $\boldsymbol{\Psi}$ if $\boldsymbol{x}$ were fully observable is given by

$$\log L_c(\boldsymbol{\Psi}) = \log g_c(\boldsymbol{x}; \boldsymbol{\Psi}).$$

The EM algorithm approaches the problem of solving the incomplete-data likelihood equation (6.1) indirectly by proceeding iteratively in terms of $\log L_c(\boldsymbol{\Psi})$. As it is unobservable, it is replaced by its conditional expectation given $\boldsymbol{y}$, using the current fit for $\boldsymbol{\Psi}$. On the $(k+1)$th iteration of the EM algorithm,
E-Step: Compute $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$, where

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}\{\log L_c(\boldsymbol{\Psi})|\boldsymbol{y}\}. \tag{6.2}$$

M-Step: Choose $\boldsymbol{\Psi}^{(k+1)}$ to be any value of $\boldsymbol{\Psi} \in \boldsymbol{\Omega}$ that maximizes $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$:

$$Q(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) \geq Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) \quad \forall \boldsymbol{\Psi} \in \boldsymbol{\Omega}. \tag{6.3}$$

The E- and M-steps are alternated repeatedly until convergence, which may be determined, for instance, by using a suitable stopping rule like $\|\boldsymbol{\Psi}^{(k+1)} - \boldsymbol{\Psi}^{(k)}\| < \varepsilon$ for some $\varepsilon > 0$ with some appropriate norm $\|\cdot\|$ or the difference $L(\boldsymbol{\Psi}^{(k+1)}) - L(\boldsymbol{\Psi}^{(k)})$ changes by an arbitrarily small amount in the case of convergence of the sequence of likelihood values $\{L(\boldsymbol{\Psi}^{(k)})\}$.

It can be shown that both the E- and M-steps will have particularly simple forms when $g_c(\boldsymbol{x}; \boldsymbol{\Psi})$ is from an exponential family:

$$g_c(\boldsymbol{x}; \boldsymbol{\Psi}) = b(\boldsymbol{x}) \exp\{\boldsymbol{c}^{\top}(\boldsymbol{\Psi})\boldsymbol{t}(\boldsymbol{x})\}/a(\boldsymbol{\Psi}), \tag{6.4}$$

where $\boldsymbol{t}(\boldsymbol{x})$ is a $k \times 1$ $(k \geq d)$ vector of complete-data sufficient statistics and $\boldsymbol{c}(\boldsymbol{\Psi})$ is a $k \times 1$ vector function of the parameter vector $\boldsymbol{\Psi}$, and $a(\boldsymbol{\Psi})$ and $b(\boldsymbol{x})$ are scalar functions. Here d is the number of unknown parameters in $\boldsymbol{\Psi}$. Members of the exponential family include most common distributions, such as the multivariate

normal, Poisson, multinomial and others. For exponential families, the E-step can be written as

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}(\log b(\boldsymbol{x})|\boldsymbol{y}) + \boldsymbol{c}^{\top}(\boldsymbol{\Psi})\boldsymbol{t}^{(k)} - \log a(\boldsymbol{\Psi}),$$

where $\boldsymbol{t}^{(k)} = E_{\boldsymbol{\Psi}^{(k)}}\{\boldsymbol{t}(\boldsymbol{X})|\boldsymbol{y}\}$ is an estimator of the sufficient statistic. The M-step maximizes the Q-function with respect to $\boldsymbol{\Psi}$; but $E_{\boldsymbol{\Psi}^{(k)}}(\log b(\boldsymbol{x})|\boldsymbol{y})$ does not depend on $\boldsymbol{\Psi}$. Hence it is sufficient to write:
E–Step: Compute

$$\boldsymbol{t}^{(k)} = E_{\boldsymbol{\Psi}^{(k)}}\{\boldsymbol{t}(\boldsymbol{X})|\boldsymbol{y}\}.$$

M–Step: Compute

$$\boldsymbol{\Psi}^{(k+1)} = \arg\max_{\boldsymbol{\Psi}}[\boldsymbol{c}^{\top}(\boldsymbol{\Psi})\boldsymbol{t}^{(k)} - \log a(\boldsymbol{\Psi})].$$

In Example 2 of Sect. 6.3.2, the complete-data p.d.f. has an exponential family representation. We shall show how the implementation of the EM algorithm can be simplified.

6.2.2 Generalized EM Algorithm

Often in practice, the solution to the M-step exists in closed form. In those instances where it does not, it may not be feasible to attempt to find the value of $\boldsymbol{\Psi}$ that globally maximizes the function $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$. For such situations, Dempster et al. (1977) defined a generalized EM (GEM) algorithm for which the M-Step requires $\boldsymbol{\Psi}^{(k+1)}$ to be chosen such that

$$Q(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) \geq Q(\boldsymbol{\Psi}^{(k)}; \boldsymbol{\Psi}^{(k)}) \tag{6.5}$$

holds. That is, one chooses $\boldsymbol{\Psi}^{(k+1)}$ to increase the Q-function, $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$, over its value at $\boldsymbol{\Psi} = \boldsymbol{\Psi}^{(k)}$, rather than to maximize it over all $\boldsymbol{\Psi} \in \boldsymbol{\Omega}$ in (6.3).

It is of interest to note that the EM (GEM) algorithm as described above implicitly defines a mapping $\boldsymbol{\Psi} \to \boldsymbol{M}(\boldsymbol{\Psi})$, from the parameter space $\boldsymbol{\Omega}$ to itself such that

$$\boldsymbol{\Psi}^{(k+1)} = \boldsymbol{M}(\boldsymbol{\Psi}^{(k)}) \qquad (k = 0, 1, 2, \ldots).$$

The function $\boldsymbol{M}$ is called the EM mapping. We shall use this function in our subsequent discussion on the convergence property of the EM algorithm.

6.2.3 Convergence of the EM Algorithm

Let $k(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{\Psi}) = g_c(\boldsymbol{x}; \boldsymbol{\Psi})/g(\boldsymbol{y}; \boldsymbol{\Psi})$ be the conditional density of $\boldsymbol{X}$ given $\boldsymbol{Y} = \boldsymbol{y}$. Then the complete-data log likelihood can be expressed by

$$\log L_c(\boldsymbol{\Psi}) = \log g_c(\boldsymbol{x}; \boldsymbol{\Psi}) = \log L(\boldsymbol{\Psi}) + \log k(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{\Psi}). \tag{6.6}$$

Taking expectations on both sides of (6.6) with respect to the conditional distribution $\boldsymbol{x}|\boldsymbol{y}$ using the fit $\boldsymbol{\Psi}^{(k)}$ for $\boldsymbol{\Psi}$, we have

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = \log L(\boldsymbol{\Psi}) + H(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}), \tag{6.7}$$

where $H(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}\{\log k(\boldsymbol{X}|\boldsymbol{y}; \boldsymbol{\Psi})|\boldsymbol{y}\}$. It follows from (6.7) that

$$\begin{aligned}\log L(\boldsymbol{\Psi}^{(k+1)}) - \log L(\boldsymbol{\Psi}^{(k)}) = \{&Q(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) - Q(\boldsymbol{\Psi}^{(k)}; \boldsymbol{\Psi}^{(k)})\} \\ &-\{H(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) - H(\boldsymbol{\Psi}^{(k)}; \boldsymbol{\Psi}^{(k)})\}. \end{aligned} \tag{6.8}$$

By Jensen's inequality and the concavity of the logarithmic function, we have $H(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) \leq H(\boldsymbol{\Psi}^{(k)}; \boldsymbol{\Psi}^{(k)})$. From (6.3) or (6.5), the first difference on the right-hand side of (6.8) is nonnegative. Hence, the likelihood function is not decreased after an EM or GEM iteration:

$$L(\boldsymbol{\Psi}^{(k+1)}) \geq L(\boldsymbol{\Psi}^{(k)}) \qquad (k = 0, 1, 2, \ldots). \tag{6.9}$$

A consequence of (6.9) is the self-consistency of the EM algorithm. Thus for a bounded sequence of likelihood values $\{L(\boldsymbol{\Psi}^{(k)})\}$, $L(\boldsymbol{\Psi}^{(k)})$ converges monotonically to some L^*. Now questions naturally arise as to the conditions under which L^* corresponds to a stationary value and when this stationary value is at least a local maximum if not a global maximum. Examples are known where the EM algorithm converges to a local *minimum* and to a saddle point of the likelihood (McLachlan and Krishnan 2008, Sect. 3.6). There are also questions of convergence of the sequence of EM iterates, that is, of the sequence of parameter values $\{\boldsymbol{\Psi}^{(k)}\}$ to the MLE.

Wu (1983) investigates in detail several convergence issues of the EM algorithm in its generality, and their relationship to other optimization methods. He shows that when the complete data are from a curved exponential family with compact parameter space, and when the Q-function satisfies a certain mild differentiability condition, then any EM sequence converges to a stationary point (not necessarily a maximum) of the likelihood function. If $L(\boldsymbol{\Psi})$ has multiple stationary points, convergence of the EM sequence to either type (local or global maximizers, saddle points) depends upon the starting value $\boldsymbol{\Psi}^{(0)}$ for $\boldsymbol{\Psi}$. If $L(\boldsymbol{\Psi})$ is unimodal in $\boldsymbol{\Omega}$ and satisfies the same differentiability condition, then any sequence $\{\boldsymbol{\Psi}^{(k)}\}$ will converge to the unique MLE of $\boldsymbol{\Psi}$, irrespective of its starting value.

To be more specific, one of the basic convergence results of the EM algorithm is the following:

$$\log L(\boldsymbol{M}(\boldsymbol{\Psi})) \geq \log L(\boldsymbol{\Psi})$$

with equality if and only if

$$Q(\boldsymbol{M}(\boldsymbol{\Psi}); \boldsymbol{\Psi}) = Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}) \quad \text{and} \quad k(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{M}(\boldsymbol{\Psi})) = k(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{\Psi}).$$

This means that the likelihood function increases at each iteration of the EM algorithm, until the condition for equality is satisfied and a fixed point of the iteration is reached. If $\hat{\boldsymbol{\Psi}}$ is an MLE, so that $\log L(\hat{\boldsymbol{\Psi}}) \geq \log L(\boldsymbol{\Psi}),\ \forall\ \boldsymbol{\Psi} \in \boldsymbol{\Omega}$, then $\log L(\boldsymbol{M}(\hat{\boldsymbol{\Psi}})) = \log L(\hat{\boldsymbol{\Psi}})$. Thus MLE are fixed points of the EM algorithm. If we have the likelihood function bounded (as might happen in many cases of interest), the EM sequence $\{\boldsymbol{\Psi}^{(k)}\}$ yields a bounded nondecreasing sequence $\{\log L(\boldsymbol{\Psi}^{(k)})\}$ which must converge as $k \to \infty$.

The theorem does not quite imply that fixed points of the EM algorithm are in fact MLEs. This is however true under fairly general conditions. For proofs and other details, see McLachlan and Krishnan (2008, Sect. 3.5) and Wu (1983). Furthermore, if a sequence of EM iterates $\{\boldsymbol{\Psi}^{(k)}\}$ satisfy the conditions

1. $[\partial Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})/\partial \boldsymbol{\Psi}]_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^{(k+1)}} = \mathbf{0}$, and
2. The sequence $\{\boldsymbol{\Psi}^{(k)}\}$ converges to some value $\boldsymbol{\Psi}^*$ and $\log k(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{\Psi})$ is sufficiently smooth,

then we have $[\partial \log L(\boldsymbol{\Psi})/\partial \boldsymbol{\Psi}]_{\boldsymbol{\Psi}=\boldsymbol{\Psi}^*} = \mathbf{0}$; see Little and Rubin (2002) and Wu (1983). Thus, despite the earlier convergence results, there is no guarantee that the convergence will be to a global maximum. For likelihood functions with multiple maxima, convergence will be to a local maximum which depends on the starting value $\boldsymbol{\Psi}^{(0)}$.

In some estimation problems with constrained parameter spaces, the parameter value maximizing the log likelihood is on the boundary of the parameter space. Here some elements of the EM sequence may lie on the boundary, thus not fulfilling Wu's conditions for convergence. Nettleton (1999) extends Wu's convergence results to the case of constrained parameter spaces and establishes some stricter conditions to guarantee convergence of the EM likelihood sequence to some local maximum and the EM parameter iterates to converge to the MLE.

6.2.4 *Rate of Convergence of the EM Algorithm*

The rate of convergence of the EM algorithm is usually slower than the quadratic convergence typically available with Newton-type methods. Dempster et al. (1977) show that the rate of convergence of the EM algorithm is linear and the rate depends on the proportion of information in the observed data. Thus in comparison to the formulated complete-data problem, if a large portion of data is missing, convergence can be quite slow.

Recall the EM mapping $\boldsymbol{M}$ defined in Sect. 6.2.2. If $\boldsymbol{\Psi}^{(k)}$ converges to some point $\boldsymbol{\Psi}^*$ and $\boldsymbol{M}(\boldsymbol{\Psi})$ is continuous, then $\boldsymbol{\Psi}^*$ is a fixed point of the algorithm; that is, $\boldsymbol{\Psi}^*$ must satisfy $\boldsymbol{\Psi}^* = \boldsymbol{M}(\boldsymbol{\Psi}^*)$. By a Taylor series expansion of $\boldsymbol{\Psi}^{(k+1)} = \boldsymbol{M}(\boldsymbol{\Psi}^{(k)})$ about the point $\boldsymbol{\Psi}^{(k)} = \boldsymbol{\Psi}^*$, we have in a neighborhood of $\boldsymbol{\Psi}^*$ that

$$\boldsymbol{\Psi}^{(k+1)} - \boldsymbol{\Psi}^* \approx \boldsymbol{J}(\boldsymbol{\Psi}^*)(\boldsymbol{\Psi}^{(k)} - \boldsymbol{\Psi}^*),$$

where $\boldsymbol{J}(\boldsymbol{\Psi})$ is the $d \times d$ Jacobian matrix for $\boldsymbol{M}(\boldsymbol{\Psi}) = (M_1(\boldsymbol{\Psi}), \ldots, M_d(\boldsymbol{\Psi}))^\top$, having (i, j)th element $r_{ij}(\boldsymbol{\Psi})$ equal to

$$r_{ij}(\boldsymbol{\Psi}) = \partial M_i(\boldsymbol{\Psi}) / \partial \Psi_j,$$

where $\Psi_j = (\boldsymbol{\Psi})_j$ and d is the dimension of $\boldsymbol{\Psi}$. Thus, in a neighborhood of $\boldsymbol{\Psi}^*$, the EM algorithm is essentially a linear iteration with rate matrix $\boldsymbol{J}(\boldsymbol{\Psi}^*)$, since $J(\boldsymbol{\Psi}^*)$ is typically nonzero. For this reason, $\boldsymbol{J}(\boldsymbol{\Psi}^*)$ is often referred to as the matrix rate of convergence. For vector $\boldsymbol{\Psi}$, a measure of the actual observed convergence rate is the global rate of convergence, which is defined as

$$r = \lim_{k \to \infty} \| \boldsymbol{\Psi}^{(k+1)} - \boldsymbol{\Psi}^* \| / \| \boldsymbol{\Psi}^{(k)} - \boldsymbol{\Psi}^* \|,$$

where $\| \cdot \|$ is any norm on d-dimensional Euclidean space $\Re^d$. It is noted that the observed rate of convergence equals the largest eigenvalue of $\boldsymbol{J}(\boldsymbol{\Psi}^*)$ under certain regularity conditions (Meng and van Dyk 1997). As a large value of r implies slow convergence, the global speed of convergence is defined to be $s = 1 - r$ (Meng 1994); see also McLachlan and Krishnan (2008, Sect. 3.9).

6.2.5 *Initialization of the EM Algorithm*

The EM algorithm will converge very slowly if a poor choice of initial value $\boldsymbol{\Psi}^{(0)}$ were used. Indeed, in some cases where the likelihood is unbounded on the edge of the parameter space, the sequence of estimates $\{\boldsymbol{\Psi}^{(k)}\}$ generated by the EM algorithm may diverge if $\boldsymbol{\Psi}^{(0)}$ is chosen too close to the boundary. Also, with applications where the likelihood equation has multiple roots corresponding to local maxima, the EM algorithm should be applied from a wide choice of starting values in any search for all local maxima. A variation of the EM algorithm (Wright and Kennedy 2000) uses interval analysis methods to locate multiple stationary points of a log likelihood within any designated region of the parameter space; see also McLachlan and Krishnan (2008, Sect. 7.9).

Different ways of specification of initial value have been considered specifically within the mixture models framework. With the EMMIX program (McLachlan and Peel 2000, pp. 343–344), an initial parameter value can be obtained automatically using either random partitions of the data, k-means clustering algorithm, or hierarchical clustering methods. With random starts, the effect of the central limit theorem tends to have the component parameters initially being similar at least in large samples. With the EMMIX program, there is an additional option for random starts to reduce this effect by first selecting a random subsample from the data, which is then randomly assigned to the g components. As described in McLachlan and Peel (2000, Sect. 2.12), the subsample has to be sufficiently large to ensure that the first M-step is able to produce a nondegenerate estimate of the parameter vector $\boldsymbol{\Psi}$.

Ueda and Nakano (1998) considered a deterministic annealing EM (DAEM) algorithm in order for the EM iterative process to be able to recover from a poor choice of starting value. They proposed using the principle of maximum entropy and the statistical mechanics analogy, whereby a parameter, say θ, is introduced with $1/\theta$ corresponding to the "temperature" in an annealing sense. With their DAEM algorithm, the E-step is effected by averaging $\log L_c(\boldsymbol{\Psi})$ over the distribution taken to be proportional to that of the current estimate of the conditional density of the complete data (given the observed data) raised to the power of θ; see for example McLachlan and Peel (2000, pp. 58–60). Recently, Pernkopf and Bouchaffra (2005) combined genetic algorithms (GA) and the EM algorithm for fitting normal mixtures, where the proposed algorithm is less sensitive to its initialization and enables escaping from local optimal solutions.

6.3 Examples of the EM Algorithm

6.3.1 Example 1: Normal Mixtures

One of the classical formulation of the statistical pattern recognition involves a mixture of p-dimensional normal distributions with a finite number, say g, of components in some unknown proportions $\pi_1, \ldots, \pi_g$ that sum to one. Here, we have n independent observations $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ from the mixture density

$$f(\boldsymbol{y}; \boldsymbol{\Psi}) = \sum_{i=1}^{g} \pi_i \phi(\boldsymbol{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where $\phi(\boldsymbol{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ denotes the p-dimensional normal density function with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$ $(i = 1, \ldots, g)$. The vector $\boldsymbol{\Psi}$ of unknown parameters consists of the mixing proportions $\pi_1, \ldots, \pi_{g-1}$, the elements of the component means $\boldsymbol{\mu}_i$, and the distinct elements of the component-covariance matrices $\boldsymbol{\Sigma}_i$. The problem of estimating $\boldsymbol{\Psi}$ is an instance of the problem of resolution of mixtures or in pattern recognition parlance an "unsupervised learning problem".

Consider the corresponding "supervised learning problem", where observations on the random vector $\boldsymbol{X} = (\boldsymbol{Z}, \boldsymbol{Y})$ are $\boldsymbol{x}_1 = (z_1, \boldsymbol{y}_1)$, $\boldsymbol{x}_2 = (z_2, \boldsymbol{y}_2), \ldots, \boldsymbol{x}_n = (z_n, \boldsymbol{y}_n)$. Here $\boldsymbol{z}_j$ is the unobservable component-indicator vector, where the ith element z_{ij} of $\boldsymbol{z}_j$ is taken to be one or zero according as the jth observation does or does not come from the ith component $(j = 1, \ldots, n)$. The MLE problem is far simpler here with easy closed-form MLE. The classificatory vectors $\boldsymbol{z} = (\boldsymbol{z}_1^\top, \ldots, \boldsymbol{z}_n^\top)^\top$ could be called the missing data. The unsupervised learning problem could be called the incomplete-data problem and the supervised learning problem the complete-data problem. A relatively simple iterative method for computing the

MLE for the unsupervised problem could be given exploiting the simplicity of the MLE for the supervised problem. This is the essence of the EM algorithm.

The complete-data log likelihood function for $\boldsymbol{\Psi}$ is given by

$$\log L_c(\boldsymbol{\Psi}) = \sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}\{\log \pi_i + \log \phi(\boldsymbol{y}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}. \tag{6.10}$$

Now the EM algorithm for this problem starts with some initial value $\boldsymbol{\Psi}^{(0)}$ for the parameters. As $\log L_c(\boldsymbol{\Psi})$ in (6.10) is a linear function of the unobservable data $\boldsymbol{z}$ for this problem, the calculation of $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$ on the E-step is effected simply by replacing z_{ij} by its current conditional expectation given the observed data $\boldsymbol{y}$, which is the usual posterior probability of the jth observation arising from the ith component

$$\tau_{ij}^{(k)} = E_{\boldsymbol{\Psi}^{(k)}}(Z_{ij}|\boldsymbol{y}) = \frac{\pi_i^{(k)}\phi(\boldsymbol{y}_j; \boldsymbol{\mu}_i^{(k)}, \boldsymbol{\Sigma}_i^{(k)})}{\sum_{l=1}^{g}\pi_l^{(k)}\phi(\boldsymbol{y}_j; \boldsymbol{\mu}_l^{(k)}, \boldsymbol{\Sigma}_l^{(k)})}.$$

From (6.10), it follows that

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = \sum_{i=1}^{g}\sum_{j=1}^{n} \tau_{ij}^{(k)}\{\log \pi_i + \log \phi(\boldsymbol{y}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\}. \tag{6.11}$$

For mixtures with normal component densities, it is computationally advantageous to work in terms of the sufficient statistics (Ng and McLachlan 2003) given by

$$\begin{aligned} T_{i1}^{(k)} &= \sum_{j=1}^{n} \tau_{ij}^{(k)} \\ \boldsymbol{T}_{i2}^{(k)} &= \sum_{j=1}^{n} \tau_{ij}^{(k)} \boldsymbol{y}_j \\ \boldsymbol{T}_{i3}^{(k)} &= \sum_{j=1}^{n} \tau_{ij}^{(k)} \boldsymbol{y}_j \boldsymbol{y}_j^T. \end{aligned} \tag{6.12}$$

By differentiating (6.11) with respect to $\boldsymbol{\Psi}$ on the basis of the sufficient statistics in (6.12), the M-step exists in closed form as

$$\begin{aligned} \pi_i^{(k+1)} &= T_{i1}^{(k)}/n \\ \boldsymbol{\mu}_i^{(k+1)} &= \boldsymbol{T}_{i2}^{(k)}/T_{i1}^{(k)} \\ \boldsymbol{\Sigma}_i^{(k+1)} &= \{\boldsymbol{T}_{i3}^{(k)} - T_{i1}^{(k)^{-1}}\boldsymbol{T}_{i2}^{(k)}\boldsymbol{T}_{i2}^{(k)^T}\}/T_{i1}^{(k)}. \end{aligned} \tag{6.13}$$

The E- and M-steps are then iterated until convergence. Unlike in the MLE for the supervised problem, in the M-step of the unsupervised problem, the posterior probabilities τ_{ij}, which are between 0 and 1, are used. The mean vectors $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ $(i = 1, \ldots, g)$ are computed using the $\tau_{ij}^{(k)}$ as weights in weighted averages.

In the case of unrestricted component-covariance matrices $\boldsymbol{\Sigma}_i$, $L(\boldsymbol{\Psi})$ is unbounded, as each data point gives rise to a singularity on the edge of the parameter space (McLachlan and Peel 2000, Sect. 3.8). In practice, the component-covariance matrices $\boldsymbol{\Sigma}_i$ can be restricted to being the same, $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$ $(i = 1, \ldots, g)$, where $\boldsymbol{\Sigma}$ is unspecified. In this case of homoscedastic normal components, the updated estimate of the common component-covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma}^{(k+1)} = \sum_{i=1}^{g} T_{i1}^{(k)} \boldsymbol{\Sigma}_i^{(k+1)} / n,$$

where $\boldsymbol{\Sigma}_i^{(k+1)}$ is given by (6.13), and the updates of π_i and $\boldsymbol{\mu}_i$ are as above in the heteroscedastic case.

6.3.2 Example 2: Censored Failure-Time Data

In survival or reliability analyses, the focus is the distribution of time T to the occurrence of some event that represents failure (for computational methods in survival analysis see also Chap. III.27). In many situations, there will be individuals who do not fail at the end of the study, or individuals who withdraw from the study before it ends. Such observations are censored, as we know only that their failure times are greater than particular values. We let $\boldsymbol{y} = (c_1, \delta_1, \ldots, c_n, \delta_n)^\top$ denote the observed failure-time data, where $\delta_j = 0$ or 1 according as the jth observation T_j is censored or uncensored at c_j $(j = 1, \ldots, n)$. That is, if T_j is uncensored, $t_j = c_j$, whereas if $t_j > c_j$, it is censored at c_j.

In the particular case where the p.d.f. for T is exponential with mean μ, we have

$$f(t; \mu) = \mu^{-1} \exp(-t/\mu) I_{(0,\infty)}(t) \qquad (\mu > 0), \tag{6.14}$$

where the indicator function $I_{(0,\infty)}(t) = 1$ for $t > 0$ and is zero elsewhere. The unknown parameter vector $\boldsymbol{\Psi}$ is now a scalar, being equal to μ. Denote by s the number of uncensored observations. By re-ordering the data so that the uncensored observations precede censored observations. It can be shown that the log likelihood function for μ is given by

$$\log L(\mu) = -s \log \mu - \sum_{j=1}^{n} c_j / \mu. \tag{6.15}$$

By equating the derivative of (6.15) to zero, the MLE of μ is

$$\hat{\mu} = \sum_{j=1}^{n} c_j / s. \tag{6.16}$$

Thus there is no need for the iterative computation of $\hat{\mu}$. But in this simple case, it is instructive to demonstrate how the EM algorithm would work and how its implementation could be simplified as the complete-data log likelihood belongs to the regular exponential family (see Sect. 6.2.1).

The complete-data vector $\boldsymbol{x}$ can be declared to be $\boldsymbol{x} = (t_1, \ldots, t_s, \boldsymbol{z}^\top)^\top$, where $\boldsymbol{z} = (t_{s+1}, \ldots, t_n)^\top$ contains the unobservable realizations of the $n - s$ censored random variables. The complete-data log likelihood is given by

$$\log L_c(\mu) = -n \log \mu - \sum_{j=1}^{n} t_j / \mu. \tag{6.17}$$

As $\log L_c(\mu)$ is a linear function of the unobservable data $\boldsymbol{z}$, the E-step is effected simply by replacing $\boldsymbol{z}$ by its current conditional expectation given $\boldsymbol{y}$. By the lack of memory of the exponential distribution, the conditional distribution of $T_j - c_j$ given that $T_j > c_j$ is still exponential with mean μ. So, we have

$$E_{\mu^{(k)}}(T_j | \boldsymbol{y}) = E_{\mu^{(k)}}(T_j | T_j > c_j) = c_j + \mu^{(k)} \tag{6.18}$$

for $j = s + 1, \ldots, n$. Accordingly, the Q-function is given by

$$Q(\mu; \mu^{(k)}) = -n \log \mu - \mu^{-1} \left\{ \sum_{j=1}^{n} c_j + (n - s)\mu^{(k)} \right\}.$$

In the M-step, we have

$$\mu^{(k+1)} = \left\{ \sum_{j=1}^{n} c_j + (n - s)\mu^{(k)} \right\} \Big/ n. \tag{6.19}$$

On putting $\mu^{(k+1)} = \mu^{(k)} = \mu^*$ in (6.19) and solving for μ^*, we have for $s < n$ that $\mu^* = \hat{\mu}$. That is, the EM sequence $\{\mu^{(k)}\}$ has the MLE $\hat{\mu}$ as its unique limit point, as $k \to \infty$; see McLachlan and Krishnan (2008, Sect. 1.5.2).

From (6.17), it can be seen that $\log L_c(\mu)$ has the exponential family form (6.4) with canonical parameter μ^{-1} and sufficient statistic $\boldsymbol{t}(\boldsymbol{X}) = \sum_{j=1}^{n} T_j$. Hence, from (6.18), the E-step requires the calculation of $\boldsymbol{t}^{(k)} = \sum_{j=1}^{n} c_j + (n - s)\mu^{(k)}$. The M-step then yields $\mu^{(k+1)}$ as the value of μ that satisfies the equation

$$\boldsymbol{t}^{(k)} = E_{\mu}\{\boldsymbol{t}(\boldsymbol{X})\} = n\mu.$$

This latter equation can be seen to be equivalent to (6.19), as derived by direct differentiation of the Q-function.

6.3.3 Example 3: Mixture-of-Experts Models

Among the various kinds of modular networks, mixtures-of-experts (Jacobs et al. 1991) and hierarchical mixtures-of-experts (Jordan and Jacobs 1994) are of much interest due to their wide applicability and the advantage of fast learning via the EM algorithm (Jordan and Xu 1995; Ng and McLachlan 2004a). In mixture-of-experts (ME) networks, there are a finite number, say m, of modules, referred to as expert networks. These expert networks approximate the distribution of the output $\boldsymbol{y}_j$ within each region of the input space. The expert network maps its input $\boldsymbol{x}_j$ to an output, the density $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)$, where $\boldsymbol{\theta}_h$ is a vector of unknown parameters for the hth expert network. It is assumed that different experts are appropriate in different regions of the input space. The gating network provides a set of scalar coefficients $\pi_h(\boldsymbol{x}_j;\boldsymbol{\alpha})$ that weight the contributions of the various experts, where $\boldsymbol{\alpha}$ is a vector of unknown parameters in the gating network. Therefore, the final output of the ME neural network is a weighted sum of all the output vectors produced by expert networks:

$$f(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\Psi}) = \sum_{h=1}^{m} \pi_h(\boldsymbol{x}_j;\boldsymbol{\alpha}) f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h), \tag{6.20}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\alpha}^{\top}, \boldsymbol{\theta}_1^{\top}, \ldots, \boldsymbol{\theta}_m^{\top})^{\top}$ is the vector of all the unknown parameters. The output of the gating network is modeled by the softmax function as

$$\pi_h(\boldsymbol{x};\boldsymbol{\alpha}) = \frac{\exp(\boldsymbol{v}_h^{\top}\boldsymbol{x})}{\sum_{l=1}^{m}\exp(\boldsymbol{v}_l^{\top}\boldsymbol{x})} \qquad (h = 1,\ldots,m), \tag{6.21}$$

where $\boldsymbol{v}_h$ is the weight vector of the hth expert in the gating network and $\boldsymbol{v}_m = \boldsymbol{0}$. It is implicitly assumed that the first element of $\boldsymbol{x}$ is one, to account for an intercept term. It follows from (6.21) that $\boldsymbol{\alpha}$ contains the elements in $\boldsymbol{v}_h$ $(h = 1,\ldots,m-1)$.

To apply the EM algorithm to the ME networks, we introduce the indicator variables z_{hj}, where z_{hj} is one or zero according to whether $\boldsymbol{y}_j$ belongs or does not belong to the hth expert (Ng and McLachlan 2004a). The complete-data log likelihood for $\boldsymbol{\Psi}$ is given by

$$\log L_c(\boldsymbol{\Psi}) = \sum_{j=1}^{n}\sum_{h=1}^{m} z_{hj}\{\log \pi_h(\boldsymbol{x}_j;\boldsymbol{\alpha}) + \log f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)\}. \tag{6.22}$$

On the $(k+1)$th iteration, the E-step calculates the Q-function as

$$\begin{aligned} Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) &= E_{\boldsymbol{\Psi}^{(k)}}\{\log L_c(\boldsymbol{\Psi})|\boldsymbol{y}, \boldsymbol{x}\} \\ &= \sum_{j=1}^{n}\sum_{h=1}^{m} E_{\boldsymbol{\Psi}^{(k)}}(Z_{hj}|\boldsymbol{y}, \boldsymbol{x})\{\log \pi_h(\boldsymbol{x}_j; \boldsymbol{\alpha}) + \log f_h(\boldsymbol{y}_j|\boldsymbol{x}_j; \boldsymbol{\theta}_h)\} \\ &= Q_\alpha + Q_\theta, \end{aligned} \tag{6.23}$$

where the Q-function can be decomposed into two terms with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}_h$ $(h = 1, \ldots, m)$, respectively, as

$$Q_\alpha = \sum_{j=1}^{n}\sum_{h=1}^{m} \tau_{hj}^{(k)} \log \pi_h(\boldsymbol{x}_j; \boldsymbol{\alpha}), \tag{6.24}$$

and

$$Q_\theta = \sum_{j=1}^{n}\sum_{h=1}^{m} \tau_{hj}^{(k)} \log f_h(\boldsymbol{y}_j|\boldsymbol{x}_j; \boldsymbol{\theta}_h), \tag{6.25}$$

where

$$\begin{aligned} \tau_{hj}^{(k)} &= E_{\boldsymbol{\Psi}^{(k)}}(Z_{hj}|\boldsymbol{y}, \boldsymbol{x}) \\ &= \pi_h(\boldsymbol{x}_j; \boldsymbol{\alpha}^{(k)}) f_h(\boldsymbol{y}_j|\boldsymbol{x}_j; \boldsymbol{\theta}_h^{(k)}) / \sum_{r=1}^{m} \pi_r(\boldsymbol{x}_j; \boldsymbol{\alpha}^{(k)}) f_r(\boldsymbol{y}_j|\boldsymbol{x}_j; \boldsymbol{\theta}_r^{(k)}) \end{aligned}$$

is the current estimated posterior probability that $\boldsymbol{y}_j$ belongs to the hth expert $(h = 1, \ldots, m)$.

Hence, the M-step consists of two separate maximization problems. With the gating network (6.21), the updated estimate of $\boldsymbol{\alpha}^{(k+1)}$ is obtained by solving

$$\sum_{j=1}^{n}\left(\tau_{hj}^{(k)} - \frac{\exp(\boldsymbol{v}_h^\top \boldsymbol{x}_j)}{1 + \sum_{l=1}^{m-1}\exp(\boldsymbol{v}_l^\top \boldsymbol{x}_j)}\right)\boldsymbol{x}_j = 0 \qquad (h = 1, \ldots, m-1), \tag{6.26}$$

which is a set of non-linear equations with $(m-1)p$ unknown parameters, where p is the dimension of $\boldsymbol{x}_j$ $(j = 1, \ldots, n)$. It can be seen from (6.26) that the non-linear equation for the hth expert depends not only on the parameter vector $\boldsymbol{v}_h$, but also on other parameter vectors $\boldsymbol{v}_l$ $(l = 1, \ldots, m-1)$. In other words, each parameter vector $\boldsymbol{v}_h$ cannot be updated independently. With the iterative reweighted least squares (IRLS) algorithm presented in Jordan and Jacobs (1994), the independence assumption on these parameter vectors was used implicitly and each parameter vector was updated independently and in parallel as

$$\boldsymbol{v}_h^{(s+1)} = \boldsymbol{v}_h^{(s)} - \gamma_\alpha \left(\frac{\partial^2 Q_\alpha}{\partial \boldsymbol{v}_h \boldsymbol{v}_h^\top} \right)^{-1} \frac{\partial Q_\alpha}{\partial \boldsymbol{v}_h} \qquad (h = 1, \ldots, m-1), \tag{6.27}$$

where $\gamma_\alpha \leq 1$ is the learning rate (Jordan and Xu 1995). That is, there are $m-1$ sets of non-linear equations each with p variables instead of a set of non-linear equations with $(m-1)p$ variables. In Jordan and Jacobs (1994), the iteration (6.27) is referred to as the inner loop of the EM algorithm. This inner loop is terminated when the algorithm has converged or the algorithm has still not converged after some pre-specified number of iterations. The above independence assumption on the parameter vectors is equivalent to the adoption of an incomplete Hessian matrix of the Q-function (Ng and McLachlan 2004a).

The densities $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)$ $(h = 1, \ldots, m)$ can be assumed to belong to the exponential family (Jordan and Jacobs 1994). In this case, the ME model (6.20) will have the form of a mixture of generalized linear models (McLachlan and Peel 2000, Sect. 5.13). The updated estimate of $\boldsymbol{\theta}_h^{(k+1)}$ is obtained by solving

$$\sum_{j=1}^{n} \tau_{hj}^{(k)} \partial \log f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)/\partial \boldsymbol{\theta}_h = \boldsymbol{0} \qquad (h = 1, \ldots, m). \tag{6.28}$$

Equation (6.28) can be solved separately for each expert $(h = 1, \ldots, m)$ when the density $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)$ is assumed to be normally distributed. With some other members of the exponential family such as multinomial distribution, (6.28) requires iterative methods to solve; see Example 5 in Sect. 6.4.2.

6.3.4 Misconceptions on the E-Step

Examples 1 to 3 may have given an impression that the E-step consists in simply replacing the missing data by their conditional expectations given the observed data at current parameter values. However, this will be valid only if the complete-data log likelihood $\log L_c(\boldsymbol{\Psi})$ were a linear function of the missing data $\boldsymbol{z}$. Unfortunately, it is not always true in general. Rather, as should be clear from the general theory described in Sect. 6.2.1, the E-step consists in replacing $\log L_c(\boldsymbol{\Psi})$ by its conditional expectation given the observed data at current parameter values. Flury and Zoppé (2000) give an example to demonstrate the point that the E-step does not always consist in plugging in "estimates" for missing data. Similar misconceptions exist in the applications of the EM algorithm to train neural networks. Let

$$(\boldsymbol{x}_1^\top, \boldsymbol{y}_1^\top)^\top, \ldots, (\boldsymbol{x}_n^\top, \boldsymbol{y}_n^\top)^\top \tag{6.29}$$

denote the n examples available for training a neural network, where $\boldsymbol{x}_j$ is an input feature vector and $\boldsymbol{y}_j$ is an output vector $(j = 1, \ldots, n)$. In the training process, the

unknown parameters in the neural network, denoted by a vector $\boldsymbol{\Psi}$, are inferred from the observed training data given by (6.29). We let $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top$ and $\boldsymbol{y} = (\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_n^\top)^\top$. In order to estimate $\boldsymbol{\Psi}$ by the statistical technique of maximum likelihood, we have to impose a statistical distribution for the observed data (6.29), which will allow us to form a log likelihood function, $\log L(\boldsymbol{\Psi}; \boldsymbol{y}, \boldsymbol{x})$, for $\boldsymbol{\Psi}$. In general, we proceed conditionally on the values for the input variable $\boldsymbol{x}$; that is, we shall consider the specification of the conditional distribution of the random variable $\boldsymbol{Y}$ corresponding to the observed output $\boldsymbol{y}$ given the input $\boldsymbol{x}$; see, for example, (6.20) in Sect. 6.3.3.

Within the EM framework, the unknown vector $\boldsymbol{\Psi}$ is estimated by consideration of the complete-data log likelihood formed on the basis of both the observed and the missing data $\boldsymbol{z}$, $\log L_c(\boldsymbol{\Psi}; \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x})$. On the $(k+1)$th iteration of the EM algorithm, the E-step computes the Q-function, which is given by

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}\{\log L_c(\boldsymbol{\Psi}; \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}) | \boldsymbol{y}, \boldsymbol{x}\}. \tag{6.30}$$

In some instances, a modified form of the EM algorithm is being used unwittingly in that on the E-step, the Q-function is effected simply by replacing the random vector $\boldsymbol{z}$ by its conditional expectation. That is, (6.30) is computed by the approximation

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) \approx \log L_c(\boldsymbol{\Psi}; \boldsymbol{y}, \tilde{\boldsymbol{z}}, \boldsymbol{x}), \tag{6.31}$$

where

$$\tilde{\boldsymbol{z}} = E_{\boldsymbol{\Psi}^{(k)}}\{\boldsymbol{Z} | \boldsymbol{y}, \boldsymbol{x}\}.$$

As described above, the approximation (6.31) will be invalid when the complete-data log likelihood is non-linear in $\boldsymbol{z}$, for example, in the multilayer perceptron networks or the radial basis function networks with regression weights; see Ng and McLachlan (2004a).

6.3.5 *Provision of Standard Errors*

Several methods have been suggested in the EM literature for augmenting the EM computation with some computation for obtaining an estimate of the covariance matrix of the computed MLE. Many such methods attempt to exploit the computations in the EM steps. These methods are based on the observed information matrix $\boldsymbol{I}(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})$, the expected information matrix $\mathcal{I}(\boldsymbol{\Psi})$ or on resampling methods. Baker (1992) reviews such methods and also develops a method for computing the observed information matrix in the case of categorical data. Jamshidian and Jennrich (2000) review more recent methods including the Supplemented EM (SEM) algorithm of Meng and Rubin (1991) and suggest some newer methods based on numerical differentiation.

Theoretically one may compute the asymptotic covariance matrix by inverting the observed or expected information matrix at the MLE. In practice, however, this

may be tedious analytically or computationally, defeating one of the advantages of the EM approach. Louis (1982) extracts the observed information matrix in terms of the conditional moments of the gradient and curvature of the complete-data log likelihood function introduced within the EM framework. These conditional moments are generally easier to work out than the corresponding derivatives of the incomplete-data log likelihood function. An alternative approach is to numerically differentiate the likelihood function to obtain the Hessian. In an EM-aided differentiation approach, Meilijson (1989) suggests perturbation of the incomplete-data score vector to compute the observed information matrix. In the SEM algorithm (Meng and Rubin 1991), numerical techniques are used to compute the derivative of the EM operator $\boldsymbol{M}$ to obtain the observed information matrix. The basic idea is to use the fact that the rate of convergence is governed by the fraction of the missing information to find the increased variability due to missing information to add to the assessed complete-data covariance matrix. More specifically, let $\boldsymbol{V}$ denote the asymptotic covariance matrix of the MLE $\hat{\boldsymbol{\Psi}}$. Meng and Rubin (1991) show that

$$\boldsymbol{I}^{-1}(\hat{\boldsymbol{\Psi}};\, \boldsymbol{y}) = \mathcal{I}_c^{-1}(\hat{\boldsymbol{\Psi}};\, \boldsymbol{y}) + \Delta \boldsymbol{V}, \tag{6.32}$$

where $\Delta \boldsymbol{V} = \{\boldsymbol{I}_d - \boldsymbol{J}(\hat{\boldsymbol{\Psi}})\}^{-1}\boldsymbol{J}(\hat{\boldsymbol{\Psi}})\mathcal{I}_c^{-1}(\hat{\boldsymbol{\Psi}};\, \boldsymbol{y})$ and $\mathcal{I}_c(\hat{\boldsymbol{\Psi}};\, \boldsymbol{y})$ is the conditional expected complete-data information matrix, and where $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix. Thus the diagonal elements of $\Delta \boldsymbol{V}$ give the increases in the asymptotic variances of the components of $\hat{\boldsymbol{\Psi}}$ due to missing data. For a wide class of problems where the complete-data density is from the regular exponential family, the evaluation of $\mathcal{I}_c(\hat{\boldsymbol{\Psi}};\, \boldsymbol{y})$ is readily facilitated by standard complete-data computations (McLachlan and Krishnan 2008, Sect. 4.5). The calculation of $\boldsymbol{J}(\hat{\boldsymbol{\Psi}})$ can be readily obtained by using only EM code via numerically differentiation of $\boldsymbol{M}(\boldsymbol{\Psi})$. Let $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}^{(k+1)}$ where the sequence of EM iterates has been stopped according to a suitable stopping rule. Let M_i be the ith component of $\boldsymbol{M}(\boldsymbol{\Psi})$. Let $\boldsymbol{u}^{(j)}$ be a column d-vector with the jth coordinate 1 and others 0. With a possibly different EM sequence $\boldsymbol{\Psi}^{(k)}$, let r_{ij} be the (i, j)th element of $\boldsymbol{J}(\hat{\boldsymbol{\Psi}})$, we have

$$r_{ij}^{(k)} = \frac{M_i[\hat{\boldsymbol{\Psi}} + (\boldsymbol{\Psi}_j^{(k)} - \hat{\boldsymbol{\Psi}}_j \boldsymbol{u}^{(j)})] - \hat{\boldsymbol{\Psi}}_i}{\boldsymbol{\Psi}_j^{(k)} - \hat{\boldsymbol{\Psi}}_j}.$$

Use a suitable stopping rule like $|r_{ij}^{(k+1)} - r_{ij}^{(k)}| < \sqrt{\epsilon}$ to stop each of the sequences r_{ij} $(i, j = 1, 2, \ldots, d)$ and take $r_{ij}^{*} = r_{ij}^{(k+1)}$; see McLachlan and Krishnan (2008, Sect. 4.5).

It is important to emphasize that estimates of the covariance matrix of the MLE based on the expected or observed information matrices are guaranteed to be valid inferentially only asymptotically. In particular for mixture models, it is well known that the sample size n has to be very large before the asymptotic theory of maximum likelihood applies. A resampling approach, the bootstrap (Efron 1979; Efron and Tibshirani 1993), has been considered to tackle this problem; see also Chernick

(2008) for recent developments of the bootstrap in statistics. Basford et al. (1997) compared the bootstrap and information-based approaches for some normal mixture models and found that unless the sample size was very large, the standard errors obtained by an information-based approach were too unstable to be recommended.

The bootstrap is a powerful technique that permits the variability in a random quantity to be assessed using just the data at hand. Standard error estimation of $\hat{\boldsymbol{\Psi}}$ may be implemented according to the bootstrap as follows. Further discussion on bootstrap and resampling methods can be found in Chaps. III.17 and III.18 of this handbook.

1. A new set of data, $\boldsymbol{y}^*$, called the bootstrap sample, is generated according to $\hat{F}$, an estimate of the distribution function of $\boldsymbol{Y}$ formed from the original observed data $\boldsymbol{y}$. That is, in the case where $\boldsymbol{y}$ contains the observed values of a random sample of size n, $\boldsymbol{y}^*$ consists of the observed values of the random sample

$$\boldsymbol{Y}_1^*, \ldots, \boldsymbol{Y}_n^* \overset{\text{i.i.d.}}{\sim} \hat{F},$$

   where the estimate $\hat{F}$ (now denoting the distribution function of a single observation $\boldsymbol{Y}_j$) is held fixed at its observed value.
2. The EM algorithm is applied to the bootstrap observed data $\boldsymbol{y}^*$ to compute the MLE for this data set, $\hat{\boldsymbol{\Psi}}^*$.
3. The bootstrap covariance matrix of $\hat{\boldsymbol{\Psi}}^*$ is given by

$$\text{Cov}^*(\hat{\boldsymbol{\Psi}}^*) = E^*[\{\hat{\boldsymbol{\Psi}}^* - E^*(\hat{\boldsymbol{\Psi}}^*)\}\{\hat{\boldsymbol{\Psi}}^* - E^*(\hat{\boldsymbol{\Psi}}^*)\}^\top], \tag{6.33}$$

   where E^* denotes expectation over the bootstrap distribution specified by $\hat{F}$.

The bootstrap covariance matrix can be approximated by Monte Carlo methods. Steps 1 and 2 are repeated independently a number of times (say, B) to give B independent realizations of $\hat{\boldsymbol{\Psi}}^*$, denoted by $\hat{\boldsymbol{\Psi}}_1^*, \ldots, \hat{\boldsymbol{\Psi}}_B^*$. Then (6.33) can be approximated by the sample covariance matrix of these B bootstrap replications to give

$$\text{Cov}^*(\hat{\boldsymbol{\Psi}}^*) \approx \sum_{b=1}^{B} (\hat{\boldsymbol{\Psi}}_b^* - \overline{\hat{\boldsymbol{\Psi}}}^*)(\hat{\boldsymbol{\Psi}}_b^* - \overline{\hat{\boldsymbol{\Psi}}}^*)^\top / (B-1), \tag{6.34}$$

where $\overline{\hat{\boldsymbol{\Psi}}}^* = \sum_{b=1}^{B} \hat{\boldsymbol{\Psi}}_b^* / B$. The standard error of the ith element of $\hat{\boldsymbol{\Psi}}$ can be estimated by the positive square root of the ith diagonal element of (6.34). It has been shown that 50 to 100 bootstrap replications are generally sufficient for standard error estimation (Efron and Tibshirani 1993).

In Step 1 above, the nonparametric version of the bootstrap would take $\hat{F}$ to be the empirical distribution function formed from the observed data $\boldsymbol{y}$. Situations where we may wish to use the latter include problems where the observed data are censored or are missing in the conventional sense.

6.4 Variations on the EM Algorithm

In this section, further modifications and extensions to the EM algorithm are considered. In general, there are extensions of the EM algorithm:

1. To produce standard errors of the MLE using the EM.
2. To surmount problems of difficult E-step and/or M-step computations.
3. To tackle problems of slow convergence.
4. In the direction of Bayesian or regularized or penalized ML estimations.

We have already discussed methods like the SEM algorithm for producing standard errors of EM-computed MLE in Sect. 6.3.5. The modification of the EM algorithm for Bayesian inference will be discussed in Sect. 6.5.1. In this section, we shall focus on the problems of complicated E- or M-steps and of slow convergence of the EM algorithm.

6.4.1 Complicated E-Step

In some applications of the EM algorithm, the E-step is complex and does not admit a close-form solution to the Q-function. In this case, the E-step at the $(k+1)$th iteration may be executed by a Monte Carlo (MC) process:

1. Make M independent draws of the missing values $\boldsymbol{Z}, \boldsymbol{z}^{(1_k)}, \ldots, \boldsymbol{z}^{(M_k)}$, from the conditional distribution $k(\boldsymbol{z}|\boldsymbol{y}; \boldsymbol{\Psi}^{(k)})$.
2. Approximate the Q-function as

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) \approx Q_M(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = \frac{1}{M} \sum_{m=1}^{M} \log k(\boldsymbol{\Psi}|\boldsymbol{z}^{(m_k)}; \boldsymbol{y}).$$

In the M-step, the Q-function is maximized over $\boldsymbol{\Psi}$ to obtain $\boldsymbol{\Psi}^{(k+1)}$. The variant is known as the Monte Carlo EM (MCEM) algorithm (Wei and Tanner 1990). As MC error is introduced at the E-step, the monotonicity property is lost. But in certain cases, the algorithm gets close to a maximizer with a high probability (Booth and Hobert 1999). The problems of specifying M and monitoring convergence are of central importance in the routine use of the algorithm (Levine and Fan 2004). Wei and Tanner (1990) recommend small values of M be used in initial stages and be increased as the algorithm moves closer to convergence. As to monitoring convergence, they recommend that the values of $\boldsymbol{\Psi}^{(k)}$ be plotted against k and when convergence is indicated by the stabilization of the process with random fluctuations about $\hat{\boldsymbol{\Psi}}$, the process may be terminated or continued with a larger value of M. Alternative schemes for specifying M and stopping rule are considered by Booth and Hobert (1999) and McCulloch (1997). The computation of standard errors with MCEM algorithm is discussed in Robert and Casella (2004, Sect. 5.3).

Example 4: Generalized Linear Mixed Models

Generalized linear mixed models (GLMM) are extensions of generalized linear models (GLM) (McCullagh and Nelder 1989) that incorporate random effects in the linear predictor of the GLM (more material on the GLM can be found in Chap. III.24). We let $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ denote the observed data vector. Conditional on the unobservable random effects vector, $\boldsymbol{u} = (u_1, \ldots, u_q)^\top$, we assume that $\boldsymbol{y}$ arise from a GLM. The conditional mean $\mu_j = E(y_j|\boldsymbol{u})$ is related to the linear predictor $\eta_j = \boldsymbol{x}_j^\top \boldsymbol{\beta} + \boldsymbol{z}_j^\top \boldsymbol{u}$ by the link function $g(\mu_j) = \eta_j$ $(j = 1, \ldots, n)$, where $\boldsymbol{\beta}$ is a p-vector of fixed effects and $\boldsymbol{x}_j$ and $\boldsymbol{z}_j$ are, respectively, p-vector and q-vector of explanatory variables associated with the fixed and random effects. This formulation encompasses the modeling of data involving multiple sources of random error, such as repeated measures within subjects and clustered data collected from some experimental units (Breslow and Clayton 1993; Ng et al. 2004).

We let the distribution for $\boldsymbol{u}$ be $g(\boldsymbol{u}; \boldsymbol{D})$ that depends on parameters $\boldsymbol{D}$. The observed data $\boldsymbol{y}$ are conditionally independent with density functions of the form

$$f(y_j|\boldsymbol{u};\, \boldsymbol{\beta}, \kappa) = \exp[m_j \kappa^{-1}\{\theta_j y_j - b(\theta_j)\} + c(y_j;\, \kappa)], \tag{6.35}$$

where θ_j is the canonical parameter, κ is the dispersion parameter, and m_j is the known prior weight. The conditional mean and canonical parameters are related through the equation $\mu_j = b'(\theta_j)$, where the prime denotes differentiation with respect to θ_j. Let $\boldsymbol{\Psi}$ denotes the vector of unknown parameters within $\boldsymbol{\beta}$, κ, and $\boldsymbol{D}$. The likelihood function for $\boldsymbol{\Psi}$ is given by

$$L(\boldsymbol{\Psi}) = \int \prod_{j=1}^{n} f(y_j|\boldsymbol{u};\, \boldsymbol{\beta}, \kappa) g(\boldsymbol{u}; \boldsymbol{D}) d\boldsymbol{u}, \tag{6.36}$$

which cannot usually be evaluated in closed form and has an intractable integral whose dimension depends on the structure of the random effects.

Within the EM framework, the random effects are considered as missing data. The complete data is then $\boldsymbol{x} = (\boldsymbol{y}^\top, \boldsymbol{u}^\top)^\top$ and the complete-data log likelihood is given by

$$\log L_c(\boldsymbol{\Psi}) = \sum_{j=1}^{n} \log f(y_j|\boldsymbol{u};\, \boldsymbol{\beta}, \kappa) + \log g(\boldsymbol{u}; \boldsymbol{D}). \tag{6.37}$$

On the $(k+1)$th iteration of the EM algorithm, the E-step involves the computation of the Q-function, $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = E_{\boldsymbol{\Psi}^{(k)}}\{\log L_c(\boldsymbol{\Psi})|\boldsymbol{y}\}$, where the expectation is with respect to the conditional distribution of $\boldsymbol{u}|\boldsymbol{y}$ with current parameter value $\boldsymbol{\Psi}^{(k)}$. As this conditional distribution involves the (marginal) likelihood function $L(\boldsymbol{\Psi})$ given in (6.36), an analytical evaluation of the Q-function for the model (6.35) will be impossible outside the normal theory mixed model (Booth and Hobert 1999). The MCEM algorithm can be adopted to tackle this problem by replacing the expectation

in the E-step with a MC approximation. Let $\boldsymbol{u}^{(1_k)}, \ldots, \boldsymbol{u}^{(M_k)}$ denote a random sample from $k(\boldsymbol{u}|\boldsymbol{y}; \boldsymbol{\Psi}^{(k)})$ at the $(k+1)$th iteration. A MC approximation of the Q-function is given by

$$Q_M(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)}) = \frac{1}{M} \sum_{m=1}^{M} \{\log f(\boldsymbol{y}|\boldsymbol{u}^{(m_k)}; \boldsymbol{\beta}, \kappa) + \log g(\boldsymbol{u}^{(m_k)}; \boldsymbol{D})\}. \quad (6.38)$$

From (6.38), it can be seen that the first term of the approximated Q-function involves only parameters $\boldsymbol{\beta}$ and κ, while the second term involves only $\boldsymbol{D}$. Thus, the maximization in the MC M-step is usually relatively simple within the GLMM context (McCulloch 1997).

Alternative simulation schemes for $\boldsymbol{u}$ can be used for (6.38). For example, Booth and Hobert (1999) proposed the rejection sampling and a multivariate t importance sampling approximations. McCulloch (1997) considered dependent MC samples using MC Newton-Raphson (MCNR) algorithm. A two-slice EM algorithm has developed by Vaida and Meng (2005) to handle GLMM with binary response, where the MC E-step is implemented via a slice sampler.

6.4.2 Complicated M-Step

One of major reasons for the popularity of the EM algorithm is that the M-step involves only complete-data ML estimation, which is often computationally simple. But if the complete-data ML estimation is rather complicated, then the EM algorithm is less attractive. In many cases, however, complete-data ML estimation is relatively simple if maximization process on the M-step is undertaken conditional on some functions of the parameters under estimation. To this end, Meng and Rubin (1993) introduce a class of GEM algorithms, which they call the Expectation–Conditional Maximization (ECM) algorithm.

ECM and Multicycle ECM Algorithms

The ECM algorithm takes advantage of the simplicity of complete-data conditional maximization by replacing a complicated M-step of the EM algorithm with several computationally simpler conditional maximization (CM) steps. Each of these CM-steps maximizes the Q-function found in the preceding E-step subject to constraints on $\boldsymbol{\Psi}$, where the collection of all constraints is such that the maximization is over the full parameter space of $\boldsymbol{\Psi}$.

A CM-step might be in closed form or it might itself require iteration, but because the CM maximizations are over smaller dimensional spaces, often they are simpler, faster, and more stable than the corresponding full maximizations called for on the M-step of the EM algorithm, especially when iteration is required. The ECM algorithm typically converges more slowly than the EM in terms of number of

iterations, but can be faster in total computer time. More importantly, the ECM algorithm preserves the appealing convergence properties of the EM algorithm, such as its monotone convergence.

We suppose that the M-step is replaced by $S > 1$ steps and let $\boldsymbol{\Psi}^{(k+s/S)}$ denote the value of $\boldsymbol{\Psi}$ on the sth CM-step of the $(k+1)$th iteration. In many applications of the ECM algorithm, the S CM-steps correspond to the situation where the parameter vector $\boldsymbol{\Psi}$ is partitioned into S subvectors,

$$\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1^\top, \ldots, \boldsymbol{\Psi}_S^\top)^\top.$$

The sth CM-step then requires the maximization of the Q-function with respect to the sth subvector $\boldsymbol{\Psi}_s$ with the other $(S-1)$ subvectors held fixed at their current values. The convergence properties and the rate of convergence of the ECM algorithm have been discussed in Meng (1994), Meng and Rubin (1993), and Sexton and Swensen (2000); see also the discussion in McLachlan and Krishnan (2008, Sect. 5.2.3), where the link to the monotone convergence of Iterative Proportional Fitting with complete data (Bishop et al. 2007, Chap. 3) is described.

It can be shown that

$$Q(\boldsymbol{\Psi}^{(k+1)}; \boldsymbol{\Psi}^{(k)}) \geq Q(\boldsymbol{\Psi}^{(k+(S-1)/S)}; \boldsymbol{\Psi}^{(k)}) \geq \cdots \geq Q(\boldsymbol{\Psi}^{(k)}; \Psi^{(k)}), \qquad (6.39)$$

which implies that the ECM algorithm is a GEM algorithm and so possesses its desirable convergence properties. As noted in Sect. 6.2.3, the inequality (6.39) is a sufficient condition for

$$L(\boldsymbol{\Psi}^{(k+1)}) \geq L(\boldsymbol{\Psi}^{(k)})$$

to hold. In many cases, the computation of an E-step may be much cheaper than the computation of the CM-steps. Hence one might wish to perform one E-step before each CM-step. A cycle is defined to be one E-step followed by one CM-step. The corresponding algorithm is called the multicycle ECM (Meng and Rubin 1993). A multicycle ECM may not necessarily be a GEM algorithm; that is, the inequality (6.39) may not be hold. However, it is not difficult to show that the multicycle ECM algorithm monotonically increases the likelihood function $L(\boldsymbol{\Psi})$ after each cycle, and hence, after each iteration. The convergence results of the ECM algorithm apply to a multicycle version of it. An obvious disadvantage of using a multicycle ECM algorithm is the extra computation at each iteration. Intuitively, as a tradeoff, one might expect it to result in larger increases in the log likelihood function per iteration since the Q-function is being updated more often (Meng 1994; Meng and Rubin 1993).

Example 5: Mixture-of-Experts Models for Multiclass Classification

It is reported in the literature that ME networks trained by the EM algorithm using the IRLS algorithm in the inner loop of the M-step often performed poorly in multiclass classification because of the incorrect independence assumption (Chen

et al. 1999); see also the discussion in Sect. 6.3.3. In this section, we present an ECM algorithm to train ME networks for multiclass classification such that the parameters in the gating and expert networks are separable. It follows that the independence assumption is not required and the parameters in both (6.26) and (6.28) can be updated separately; see, for example, Ng and McLachlan (2004a) and Ng et al. (2006a).

For multiclass classification, the densities $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\boldsymbol{\theta}_h)$ $(h = 1,\ldots,m)$ are modelled by a multinomial distribution consisting of one draw on multiple (say, g) categories. That is, we have

$$f_h(\boldsymbol{y}_j|\boldsymbol{x}_j,\boldsymbol{\theta}_h) = \prod_{i=1}^{g-1}\left(\frac{\exp(\boldsymbol{w}_{hi}^\top \boldsymbol{x}_j)}{1+\sum_{r=1}^{g-1}\exp(\boldsymbol{w}_{hr}^\top \boldsymbol{x}_j)}\right)^{y_{ij}}\left(\frac{1}{1+\sum_{r=1}^{g-1}\exp(\boldsymbol{w}_{hr}^\top \boldsymbol{x}_j)}\right)^{y_{gj}}, \tag{6.40}$$

where $\boldsymbol{\theta}_h$ contains the elements in $\boldsymbol{w}_{hi}$ $(i = 1,\ldots,g-1)$. Equation (6.28) in Sect. 6.3.3 thus becomes

$$\sum_{j=1}^{n}\tau_{hj}^{(k)}\left(y_{ij} - \frac{\exp(\boldsymbol{w}_{hi}^\top \boldsymbol{x}_j)}{1+\sum_{r=1}^{g-1}\exp(\boldsymbol{w}_{hr}^\top \boldsymbol{x}_j)}\right)\boldsymbol{x}_j = \boldsymbol{0} \qquad (i = 1,\ldots,g-1) \tag{6.41}$$

for $h = 1,\ldots,m$, which are m sets of non-linear equations each with $(g-1)p$ unknown parameters.

With the ECM algorithm, the M-step is replaced by several computationally simpler CM-steps. For example, the parameter vector $\boldsymbol{\alpha}$ is partitioned as $(\boldsymbol{v}_1^\top,\ldots,\boldsymbol{v}_{m-1}^\top)^\top$. On the $(k+1)$th iteration of the ECM algorithm, the E-step is the same as given in Equations (6.23)–(6.25) for the EM algorithm, but the M-step of the latter is replaced by $(m-1)$ CM-steps, as follows:

- *CM-step 1:* Calculate $\boldsymbol{v}_1^{(k+1)}$ by maximizing Q_α with $\boldsymbol{v}_l$ $(l = 2,\ldots,m-1)$ fixed at $\boldsymbol{v}_l^{(k)}$.
- *CM-step 2:* Calculate $\boldsymbol{v}_2^{(k+1)}$ by maximizing Q_α with $\boldsymbol{v}_1$ fixed at $\boldsymbol{v}_1^{(k+1)}$ and $\boldsymbol{v}_l$ $(l = 3,\ldots,m-1)$ fixed at $\boldsymbol{v}_l^{(k)}$.
- $\vdots$
- *CM-step* $(\boldsymbol{m-1})$ **:** Calculate $\boldsymbol{v}_{(m-1)}^{(k+1)}$ by maximizing Q_α with $\boldsymbol{v}_l$ $(l = 1,\ldots,m-2)$ fixed at $\boldsymbol{v}_l^{(k+1)}$.

As each CM-step above corresponds to a separable set of the parameters in $\boldsymbol{v}_h$ for $h = 1,\ldots,m-1$, it can be obtained using the IRLS approach; see Ng and McLachlan (2004a).

6.4.3 Speeding Up Convergence

Several suggestions are available in the literature for speeding up convergence, some of a general kind and some problem-specific; see for example McLachlan

and Krishnan (2008, Chap. 4). Most of them are based on standard numerical analytic methods and suggest a hybrid of EM with methods based on Aitken acceleration, over-relaxation, line searches, Newton methods, conjugate gradients, etc. Unfortunately, the general behaviour of these hybrids is not always clear and they may not yield monotonic increases in the log likelihood over iterations. There are also methods that approach the problem of speeding up convergence in terms of "efficient" data augmentation scheme (Meng and van Dyk 1997). Since the convergence rate of the EM algorithm increases with the proportion of observed information in the prescribed EM framework (Sect. 6.2.4), the basic idea of the scheme is to search for an efficient way of augmenting the observed data. By efficient, they mean less augmentation of the observed data (greater speed of convergence) while maintaining the simplicity and stability of the EM algorithm. A common trade-off is that the resulting E- and/or M-steps may be made appreciably more difficult to implement. To this end, Meng and van Dyk (1997) introduce a working parameter in their specification of the complete data to index a class of possible schemes to facilitate the search.

ECME, AECM, and PX–EM Algorithms

Liu and Rubin (1994, 1998) present an extension of the ECM algorithm called the ECME (expectation–conditional maximization either) algorithm. Here the "either" refers to the fact that with this extension, each CM-step either maximizes the Q-function or the actual (incomplete-data) log likelihood function $\log L(\boldsymbol{\Psi})$, subject to the same constraints on $\boldsymbol{\Psi}$. The latter choice should lead to faster convergence as no augmentation is involved. Typically, the ECME algorithm is more tedious to code than the ECM algorithm, but the reward of faster convergence is often worthwhile especially because it allows convergence to be more easily assessed.

A further extension of the EM algorithm, called the Space-Alternating Generalized EM (SAGE), has been proposed by Fessler and Hero (1994), where they update sequentially small subsets of parameters using appropriately smaller complete data spaces. This approach is eminently suitable for situations like image reconstruction where the parameters are large in number. Meng and van Dyk (1997) combined the ECME and SAGE algorithms. The so-called Alternating ECM (AECM) algorithm allows the data augmentation scheme to vary where necessary over the CM-steps, within and between iterations. With this flexible data augmentation and model reduction schemes, the amount of data augmentation decreases and hence efficient computations are achieved.

In contrast to the AECM algorithm where the optimal value of the working parameter is determined before EM iterations, a variant is considered by Liu et al. (1998) which maximizes the complete-data log likelihood as a function of the working parameter within each EM iteration. The so-called parameter-expanded EM (PX–EM) algorithm has been used for fast stable computation of MLE in a wide range of models (Little and Rubin 2002). This variant has been further developed, known as the one-step-late PX–EM algorithm, to compute maximum *a posteriori*

(MAP) or maximum penalized likelihood (MPL) estimates (van Dyk and Tang 2003). Analogous convergence results hold for the ECME, AECM, and PX–EM algorithms as for the EM and ECM algorithms. More importantly, these algorithms preserve the monotone convergence of the EM algorithm.

Incremental Scheme of the EM Algorithm

The EM algorithm can be viewed as alternating minimization of a joint function between a parameter space $\boldsymbol{\Omega}$ and a family of distributions $\boldsymbol{\Phi}$ over the unobserved variables (Csiszár and Tusnády 1984; Hathaway 1986). Let $\boldsymbol{z}$ denote the vector containing the unobservable data and let P be any distribution defined over the support of $\boldsymbol{Z}$. The joint function is defined as

$$D(P, \boldsymbol{\Psi}) = -\log L(\boldsymbol{\Psi}) + KL[P, g(\boldsymbol{z}|\boldsymbol{y}; \boldsymbol{\Psi})], \tag{6.42}$$

where $g(\boldsymbol{z}|\boldsymbol{y}; \boldsymbol{\Psi})$ is the conditional distribution of $\boldsymbol{Z}$ given the observed data and $KL[P, g(\boldsymbol{z}|\boldsymbol{y}; \boldsymbol{\Psi})]$ is the Kullback-Leibler information that measures the divergence of P relative to $g(\boldsymbol{z}|\boldsymbol{y}; \boldsymbol{\Psi})$. Hathaway (1986) shows that, given the current estimates $\boldsymbol{\Psi}^{(k)}$, the E-step on the $(k+1)$th scan corresponds to the minimization of (6.42) with respect to P over $\boldsymbol{\Phi}$. For fixed $P^{(k+1)}$, the M-step then minimizes (6.42) with respect to $\boldsymbol{\Psi}$ over $\boldsymbol{\Omega}$.

From this perspective, Neal and Hinton (1998) justify an incremental variant of the EM algorithm in which only a block of unobserved data is calculated in each E-step at a time before performing a M-step. A scan of the incremental EM (IEM) algorithm thus consists of B "partial" E-steps and B M-steps, where B is the total number of blocks of data. This variant of the EM algorithm has been shown empirically to give faster convergence compared to the EM algorithm in applications where the M-step is computationally simple, for example, in fitting multivariate normal mixtures (Ng and McLachlan 2003, 2004b). With the IEM algorithm, Neal and Hinton (1998) showed that the partial E-step and the M-step both monotonically increase $F(P, \boldsymbol{\Psi}) = -D(P, \boldsymbol{\Psi})$ and if a local maximum (or saddle point) of $F(P, \boldsymbol{\Psi})$ occurs at P^* and $\boldsymbol{\Psi}^*$, then a local maximum (or saddle point) of the log likelihood occurs at $\boldsymbol{\Psi}^*$ as well. Although the IEM algorithm can possess stable convergence to stationary points in the log likelihood under slightly stronger conditions of Wu (1983) for the EM algorithm, the current theoretical results for the IEM algorithm do not quarantine monotonic behaviour of the log likelihood as the EM algorithm does. The same argument for proving that the EM algorithm always increases the log likelihood cannot be adopted here, as the estimate of $\boldsymbol{\Psi}$ in $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(k)})$ of (6.7) is changing at each iteration within each scan (Ng and McLachlan 2003). However, it is noted that $F(P, \boldsymbol{\Psi})$ can be considered as a lower bound on the log likelihood since the Kullback-Leibler information is non-negative. For given P, as obtained in the partial E-step, the M-step increases $F(P, \boldsymbol{\Psi})$ with respect to $\boldsymbol{\Psi}$. It follows that

$$F(P, \boldsymbol{\Psi}^{(k+(b+1)/B)}) \geq F(P, \boldsymbol{\Psi}^{(k+b/B)}) \qquad (b = 0, \ldots, B\text{-}1).$$

That is, the lower bound of the log likelihood is monotonic increasing after each iteration.

The argument for improved rate of convergence is that the IEM algorithm exploits new information more quickly rather than waiting for a complete scan of the data before parameters are updated by an M-step. Another method suggested by Neal and Hinton (1998) is the sparse EM (SPEM) algorithm. In fitting a mixture model to a data set by ML via the EM, the current estimates of some posterior probabilities $\tau_{ij}^{(k)}$ for a given data point $\boldsymbol{y}_j$ are often close to zero. For example, if $\tau_{ij}^{(k)} < 0.005$ for the first two components of a four-component mixture being fitted, then with the SPEM algorithm we would fix $\tau_{ij}^{(k)}$ (i=1,2) for membership of $\boldsymbol{y}_j$ with respect to the first two components at their current values and only update $\tau_{ij}^{(k)}$ (i=3,4) for the last two components. This sparse E-step will take time proportional to the number of components that needed to be updated. A sparse version of the IEM algorithm (SPIEM) can be formulated by combining the partial E-step and the sparse E-step. With these versions, the likelihood is still found to be increased after each scan. Ng and McLachlan (2003) study the relative performances of these algorithms with various number of blocks B for the fitting of normal mixtures. They propose to choose B to be that factor of n that is the closest to $B^* = \text{round}(n^{2/5})$ for unrestricted component-covariance matrices, where round(r) rounds r to the nearest integer.

Ng and McLachlan (2004b) propose to speed up further the IEM and SPIEM algorithms for the fitting of normal mixtures by imposing a multiresolution kd-tree (mrkd-tree) structure in performing the E-step. Here kd stands for k-dimensional where, in our notation, $k = p$, the dimension of an observation $\boldsymbol{y}_j$. The mrkd-tree is a binary tree that recursively splits the whole set of data points into partition (Moore 1999). The contribution of all the data points in a tree node to the sufficient statistics is simplified by calculating at the mean of these data points to save time. The mrkd-tree approach does not guarantee the desirable reliable convergence properties of the EM algorithm. However, the IEM-based mrkd-tree algorithms have been shown empirically to give a monotonic convergence as reliable as the EM algorithm when the size of leaf nodes are sufficiently small (Ng and McLachlan 2004b). It is noted that the number of leaf nodes will increase dramatically when the dimension of the data points p increases. This implies that mrkd-trees-based algorithms will not be able to speed up the EM algorithm for applications to high dimensional data (Ng and McLachlan 2004b). Recently, a number of techniques have been developed to reduce dimensionality without losing significant information and separability among mixture components; see, for example, the matrix factorization approach of Nikulin and McLachlan (2010) and the references therein.

6.5 Miscellaneous Topics on the EM Algorithm

6.5.1 EM Algorithm for MAP Estimation

Although we have focussed on the application of the EM algorithm for computing MLEs in a frequentist framework, it can be equally applied to find the mode of the posterior distribution in a Bayesian framework. This problem is analogous to MLE and hence the EM algorithm and its variants can be adapted to compute maximum *a posteriori* (MAP) estimates. The computation of the MAP estimate in a Bayesian framework via the EM algorithm corresponds to the consideration of some prior density for $\boldsymbol{\Psi}$. The E-step is effectively the same as for the computation of the MLE of $\boldsymbol{\Psi}$ in a frequentist framework, requiring the calculation of the Q-function. The M-step differs in that the objective function for the maximization process is equal to the Q-function, augmented by the log prior density. The combination of prior and sample information provides a posterior distribution of the parameter on which the estimation is based.

The advent of inexpensive high speed computers and the simultaneous rapid development in posterior simulation techniques such as Markov chain Monte Carlo (MCMC) methods (Gelfand and Smith 1990) enable Bayesian estimation to be undertaken. In particular, posterior quantities of interest can be approximated through the use of MCMC methods such as the Gibbs sampler. Such methods allow the construction of an ergodic Markov chain with stationary distribution equal to the posterior distribution of the parameter of interest. A concise theoretical treatment of MCMC is provided in Gamerman and Lopes (2006) and Robert and Casella (2004); see also McLachlan and Krishnan (2008, Chap. 8) and the references therein. A detailed description of the MCMC technology can also be found in Chap. II.4.

Although the application of MCMC methods is now routine, there are some difficulties that have to be addressed with the Bayesian approach, particularly in the context of mixture models. One main hindrance is that improper priors yield improper posterior distributions. Another hindrance is that when the number of components g is unknown, the parameter space is simultaneously ill-defined and of infinite dimension. This prevents the use of classical testing procedures and priors (McLachlan and Peel 2000, Chap. 4).

6.5.2 Iterative Simulation Algorithms

In computing Bayesian solutions to incomplete-data problems, iterative simulation techniques have been adopted to find the MAP estimates or estimating the entire posterior density. These iterative simulation techniques are conceptually similar to the EM algorithm, simply replacing the E- and M-steps by draws from the current conditional distribution of the missing data and $\boldsymbol{\Psi}$, respectively. However, in some methods such as the MCEM algorithm described in Sect. 6.4.1, only the E-step is

so implemented. Many of these methods can be interpreted as iterative simulation analogs of the various versions of the EM and its extensions. Some examples are Stochastic EM, Data Augmentation algorithm, and MCMC methods such as the Gibbs sampler (McLachlan and Krishnan 2008, Chap. 6). Here, we give a very brief outline of the Gibbs sampler; see also Chap. II.4 of this handbook and the references therein.

The Gibbs sampler is extensively used in many Bayesian problems where the joint distribution is too complicated to handle, but the conditional distributions are often easy enough to draw from; see Casella and George (1992). On the Gibbs sampler, an approximate sample from $p(\boldsymbol{\Psi} \mid \boldsymbol{y})$ is obtained by simulating directly from the (full) conditional distribution of a subvector of $\boldsymbol{\Psi}$ given all the other parameters in $\boldsymbol{\Psi}$ and $\boldsymbol{y}$. We write $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_d)$ in component form, a d-dimensional Gibbs sampler makes a Markov transition from $\boldsymbol{\Psi}^{(k)}$ to $\boldsymbol{\Psi}^{(k+1)}$ via d successive simulations as follows:

(1) Draw $\Psi_1^{(k+1)}$ from $p(\Psi_1 \mid \boldsymbol{y}; \Psi_2^{(k)}, \ldots, \Psi_d^{(k)})$.
(2) Draw $\Psi_2^{(k+1)}$ from $p(\Psi_2 \mid \boldsymbol{y}; \Psi_1^{(k+1)}, \Psi_3^{(k)} \ldots, \Psi_d^{(k)})$.
$\vdots$ $\vdots$ $\vdots$
(d) Draw $\Psi_d^{(k+1)}$ from $p(\Psi_d \mid \boldsymbol{y}; \Psi_1^{(k+1)}, \ldots, \Psi_{d-1}^{(k+1)})$.

The vector sequence $\{\boldsymbol{\Psi}^{(k)}\}$ thus generated is known to be a realization of a homogeneous Markov Chain. Many interesting properties of such a Markov sequence have been established, including geometric convergence, as $k \to \infty$; to a unique stationary distribution that is the posterior density $p(\Psi_1^{(k)}, \ldots, \Psi_d^{(k)} \mid \boldsymbol{y})$ under certain conditions; see Roberts and Polson (1994). Among other sampling methods, there is the Metropolis-Hastings algorithm (Hastings 1970), which, in contrast to the Gibbs sampler, accepts the candidate simulated component in $\boldsymbol{\Psi}$ with some defined probability (McLachlan and Peel 2000, Chap. 4).

The Gibbs sampler and other such iterative simulation techniques being Bayesian in their point of view consider both parameters and missing values as random variables and both are subjected to random draw operations. In the iterative algorithms under a frequentist framework, like the EM-type algorithms, parameters are subjected to a maximization operation and missing values are subjected to an averaging operation. Thus the various versions of the Gibbs sampler can be viewed as stochastic analogs of the EM, ECM, and ECME algorithms (Robert and Casella 2004). Besides these connections, the EM-type algorithms also come in useful as starting points for iterative simulation algorithms where typically regions of high density are not known *a priori* (McLachlan and Krishnan 2008, Sect. 6.10). The relationship between the EM algorithm and the Gibbs sampler and the connection between their convergence properties have been examined in Sahu and Roberts (1999).

6.5.3 Further Applications of the EM Algorithm

Since the publication of Dempster et al. (1977), the number, variety, and range of applications of the EM algorithm and its extensions have been tremendous. Applications in many different contexts can be found in monographs Little and Rubin (2002), McLachlan et al. (2004), McLachlan and Krishnan (2008), and McLachlan and Peel (2000). We conclude the chapter with a quick summary of some of the more interesting and topical applications of the EM algorithm.

Bioinformatics: EMMIX-GENE and EMMIX-WIRE Procedures

In bioinformatics, much attention is centered on the cluster analysis of the tissue samples and also the genes. The clustering of tumour tisses can play a useful role in the discovery and understanding of new subtypes of diseases (McLachlan et al. 2002), while the clustering of gene expression profiles contributes significantly to the elucidation of unknown gene function, the validation of gene discoveries and the interpretation of biological processes (Ng et al. 2006b). The EM algorithm and its variants have been applied to tackle some of the problems arisen in such applications. For example, the clustering of tumour tissues on the basis of genes expression is a nonstandard cluster analysis problem since the dimension of each tissue sample is so much greater than the number of tissues. The EMMIX-GENE procedure of McLachlan et al. (2002) handles the problem of a high-dimensional feature vector by using mixtures of factor analyzers whereby the component correlations between the genes are explained by their conditional linear dependence on a small number of latent or unobservable factors specific to each component. The mixtures of factor analyzers model can be fitted by using the AECM algorithm (Meng and van Dyk 1997); see, for example, McLachlan et al. (2004).

The clustering of gene profiles is also not straightforward as the profiles of the genes are not all independently distributed and the expression levels may have been obtained from an experimental design involving replicated arrays (Lee et al. 2000; Pavlidis et al. 2003). Similarly, in time-course studies (Storey et al. 2005), where expression levels are measured under various conditions or at different time points, gene expressions obtained from the same condition (tissue sample) are correlated. Ng et al. (2006b) have developed a random-effects model that provides a unified approach to the clustering of genes with correlated expression levels measured in a wide variety of experimental situations. The EMMIX-WIRE procedure of Ng et al. (2006b) formulates a linear-mixed-effects model (LMM) for the mixture components in which both gene-specific and tissue-specific random effects are incorporated in the modelling of the microarray data. In their model, the gene profiles are not all independently distributed as genes within the same component in the mixture model are allowed to be dependent due to the presence of the tissue-specific random effects. This problem is circumvented by proceeding conditionally on the tissue-specific random effects, as given these terms, the gene

profiles are all conditionally independent. In this way, Ng et al. (2006b) showed that the unknown parameter vector $\boldsymbol{\Psi}$ can be estimated by ML via the EM algorithm under a conditional mode, where both the E- and M-steps are carried out in closed form.

Health Science: On-Line Prediction of Hospital Resource Utilization

The continuing development and innovative use of information technology in health care has played a significant role in contributing and advancing this active and burgeoning field. Inpatient length of stay (LOS) is an important measure of hospital activity and health care utilization. It is also considered to be a measurement of disease severity and patient acuity (Ng et al. 2006a; Pofahl et al. 1998). Length of stay predictions have therefore important implications in various aspects of health care decision support systems. Most prediction tools use a batch-mode training process. That is, the model is trained only after the entire training set is available. Such a training method is unrealistic in the prediction of LOS as the data become available over time and the input-output pattern of data changes dynamically over time.

An intelligent ME network for on-line prediction of LOS via an incremental ECM algorithm has been proposed by Ng et al. (2006a). The strength of an incremental training process is that it enables the network to be updated when an input-output datum becomes known. These on-line and incremental updating features increase the simulation between neural networks and human decision making capability in terms of learning from "every" experience. In addition, an on-line process is capable of providing an output whenever a new datum becomes available. This on-the-spot information is therefore more useful and practical for adaptive training of model parameters and making decisions (Jepson et al. 2003; Lai and Fang 2005), especially when one deals with a tremendous amount of data.

The incremental training process for on-line prediction is formulated based on the incremental scheme of the EM algorithm described in Sect. 6.4.3; see also Ng and McLachlan (2003) and Ng et al. (2006a). In particular, the unknown parameters are updated in the CM-step when a single input-output datum is available. Also, a discount parameter is introduced to gradually "forget" the effect of previous estimated posterior probabilities obtained from earlier less-accurate estimates (Jordan and Jacobs 1994; Sato and Ishii 2000). It implies that the sufficient statistics required in the CM-step are decayed exponentially with a multiplicative discount factor as the training proceeds. When the discount parameter is scheduled to approach one as the iteration tends to infinity, the updating rules so formed can be considered as a stochastic approximation for obtaining the ML estimators (Sato and Ishii 2000; Titterington 1984).

References

Baker, S.G.: A simple method for computing the observed information matrix when using the EM algorithm with categorical data. J. Comput. Graph. Stat. **1**, 63–76 (1992)

Basford, K.E., Greenway, D.R., McLachlan, G.J., Peel, D.: Standard errors of fitted means under normal mixture models. Comput. Stat. **12**, 1–17 (1997)

Bishop, Y.M.M., Fienberg, S.E., Holland, P.W.: Discrete Multivariate Analysis: Theory and Practice. Springer, New York (2007)

Booth, J.G., Hobert, J.P.: Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. J. Roy. Stat. Soc. B **61**, 265–285 (1999)

Breslow, N.E., Clayton, D.G.: Approximate inference in generalized linear mixed models. J. Am. Stat. Assoc. **88**, 9–25 (1993)

Casella, G., George, E.I.: Explaining the Gibbs sampler. Am. Stat. **46**, 167–174 (1992)

Chen, K., Xu, L., Chi, H.: Improved learning algorithms for mixture of experts in multiclass classification. Neural Netw. **12**, 1229–1252 (1999)

Chernick, M.R.: Bootstrap Methods: A Guide for Practitioners and Researchers. Wiley, Hoboken, New Jersey (2008)

Cramér, H.: Mathematical Methods of Statistics. Princeton University Press, Princeton, New Jersey (1946)

Csiszár, I., Tusnády, G.: Information geometry and alternating minimization procedure. In: Dudewicz, E.J., Plachky, D., Sen, P.K. (eds.) Recent Results in Estimation Theory and Related Topics, pp. 205–237. R. Oldenbourg, Munich (1984)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B **39**, 1–38 (1977)

Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**, 1–26 (1979)

Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, London (1993)

Fessler, J.A., Hero, A.O.: Space-alternating generalized expectation-maximization algorithm. IEEE Trans. Signal. Process. **42**, 2664–2677 (1994)

Flury, B., Zoppé, A.: Exercises in EM. Am. Stat. **54**, 207–209 (2000)

Gamerman, D., Lopes, H.F.: Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, 2nd edn. Chapman & Hall/CRC, Boca Raton, FL (2006)

Gelfand, A.E., Smith, A.F.M.: Sampling-based approaches to calculating marginal densities. J. Am. Stat. Assoc. **85**, 398–409 (1990)

Hathaway, R.J.: Another interpretation of the EM algorithm for mixture distributions. Stat. Probab. Lett. **4**, 53–56 (1986)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109 (1970)

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**, 79–87 (1991)

Jamshidian, M., Jennrich, R.I.: Standard errors for EM estimation. J. Roy. Stat. Soc. B **62**, 257–270 (2000)

Jepson, A.D., Fleet, D.J., EI-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 1296–1311 (2003)

Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Comput. **6**, 181–214 (1994)

Jordan, M.I., Xu, L.: Convergence results for the EM approach to mixtures of experts architectures. Neural Netw. **8**, 1409–1431 (1995)

Lai, S.H., Fang, M.: An adaptive window width/center adjustment system with online training capabilities for MR images. Artif. Intell. Med. **33**, 89–101 (2005)

Lee, M.L.T., Kuo, F.C., Whitmore, G.A., Sklar, J.: Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. Proc. Natl. Acad. Sci. USA **97**, 9834–9838 (2000)

Levine, R., Fan, J.J.: An automated (Markov chain) Monte Carlo EM algorithm. J. Stat. Comput. Simulat. **74**, 349–359 (2004)
Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data, 2nd edn. Wiley, New York (2002)
Liu, C., Rubin, D.B.: The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. Biometrika **81**, 633–648 (1994)
Liu, C., Rubin, D.B.: Maximum likelihood estimation of factor analysis using the ECME algorithm with complete and incomplete data. Stat. Sin. **8**, 729–747 (1998)
Liu, C., Rubin, D.B., Wu, Y.N.: Parameter expansion to accelerate EM: the PX–EM algorithm. Biometrika **85**, 755–770 (1998)
Louis, T.A.: Finding the observed information matrix when using the EM algorithm. J. Roy. Stat. Soc. B **44**, 226–233 (1982)
McCullagh, P.A., Nelder, J.: Generalized Linear Models, 2nd edn. Chapman & Hall, London (1989)
McCulloch, C.E.: Maximum likelihood algorithms for generalized linear mixed models. J. Am. Stat. Assoc. **92**, 162–170 (1997)
McLachlan, G.J., Basford, K.E.: Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York (1988)
McLachlan, G.J., Bean, R.W., Peel, D.: A mixture model-based approach to the clustering of microarray expression data. Bioinformatics **18**, 413–422 (2002)
McLachlan, G.J., Do, K.A., Ambroise, C.: Analyzing Microarray Gene Expression Data. Wiley, New York (2004)
McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, 2nd edn. Wiley, Hoboken, New Jersey (2008)
McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
Meilijson, I.: A fast improvement of the EM algorithm in its own terms. J. Roy. Stat. Soc. B **51**, 127–138 (1989)
Meng, X.L.: On the rate of convergence of the ECM algorithm. Ann. Stat. **22**, 326–339 (1994)
Meng, X.L., Rubin, D.B.: Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm. J. Am. Stat. Assoc. **86**, 899–909 (1991)
Meng, X.L., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika **80**, 267–278 (1993)
Meng, X.L., van Dyk, D.: The EM algorithm – an old folk song sung to a fast new tune. J. Roy. Stat. Soc. B **59**, 511–567 (1997)
Moore, A.W.: Very fast EM-based mixture model clustering using multiresolution kd-trees. In: Kearns, M.S., Solla, S.A., Cohn, D.A. (eds.) Advances in Neural Information Processing Systems 11, pp. 543–549. MIT Press, MA (1999)
Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 355–368. Kluwer, Dordrecht (1998)
Nettleton, D.: Convergence properties of the EM algorithm in constrained parameter spaces. Can. J. Stat. **27**, 639–648 (1999)
Ng, S.K., McLachlan, G.J.: On the choice of the number of blocks with the incremental EM algorithm for the fitting of normal mixtures. Stat. Comput. **13**, 45–55 (2003)
Ng, S.K., McLachlan, G.J (2004a). Using the EM algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. IEEE Trans. Neural Netw. **15**, 738–749.
Ng, S.K., McLachlan, G.J (2004b). Speeding up the EM algorithm for mixture model-based segmentation of magnetic resonance images. Pattern Recogn. **37**, 1573–1589.
Ng, S.K., McLachlan, G.J., Lee, A.H (2006a). An incremental EM-based learning approach for on-line prediction of hospital resource utilization. Artif. Intell. Med. **36**, 257–267.
Ng, S.K., McLachlan, G.J., Wang, K., Ben-Tovim Jones, L., Ng, S.W (2006b). A mixture model with random-effects components for clustering correlated gene-expression profiles. Bioinformatics **22**, 1745–1752.

Ng, S.K., McLachlan, G.J., Yau, K.K.W., Lee, A.H.: Modelling the distribution of ischaemic stroke-specific survival time using an EM-based mixture approach with random effects adjustment. Stat. Med. **23**, 2729–2744 (2004)

Nikulin, V., McLachlan, G.J.: A gradient-based algorithm for matrix factorization applied to dimensionality reduction. In: Fred, A., Filipe, J., Gamboa, H. (eds.) Proceedings of BIOSTEC 2010, the 3rd International Joint Conference on Biomedical Engineering Systems and Technologies, pp. 147–152. Institute for Systems and Technologies of Information, Control and Communication, Portugal (2010)

Pavlidis, P., Li, Q., Noble, W.S.: The effect of replication on gene expression microarray experiments. Bioinformatics **19**, 1620–1627 (2003)

Pernkopf, F., Bouchaffra, D.: Genetic-based EM algorithm for learning Gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1344–1348 (2005)

Pofahl, W.E., Walczak, S.M., Rhone, E., Izenberg, S.D.: Use of an artificial neural network to predict length of stay in acute pancreatitis. Am. Surg. **64**, 868–872 (1998)

Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer, New York (2004)

Roberts, G.O., Polson, N.G.: On the geometric convergence of the Gibbs sampler. J. Roy. Stat. Soc. B **56**, 377–384 (1994)

Sahu, S.K., Roberts, G.O.: On convergence of the EM algorithm and the Gibbs sampler. Stat. Comput. **9**, 55–64 (1999)

Sato, M., Ishii, S.: On-line EM algorithm for the normalized Gaussian network. Neural Comput. **12**, 407–432 (2000)

Sexton, J., Swensen, A.R.: ECM algorithms that converge at the rate of EM. Biometrika **87**, 651–662 (2000)

Storey, J.D., Xiao, W., Leek, J.T., Tompkins, R.G., Davis, R.W.: Significance analysis of time course microarray experiments. Proc. Natl. Acad. Sci. USA **102**, 12837–12842 (2005)

Titterington, D.M.: Recursive parameter estimation using incomplete data. J. Roy. Stat. Soc. B **46**, 257–267 (1984)

Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. Neural Netw. **11**, 271–282 (1998)

van Dyk, D.A., Tang, R.: The one-step-late PXEM algorithm. Stat. Comput. **13**, 137–152 (2003)

Vaida, F., Meng, X.L.: Two-slice EM algorithms for fitting generalized linear mixed models with binary response. Stat. Modelling **5**, 229–242 (2005)

Wei, G.C.G., Tanner, M.A.: A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. J. Am. Stat. Assoc. **85**, 699–704 (1990)

Wright, K., Kennedy, W.J.: An interval analysis approach to the EM algorithm. J. Comput. Graph. Stat. **9**, 303–318 (2000)

Wu, C.F.J.: On the convergence properties of the EM algorithm. Ann. Stat. **11**, 95–103 (1983)