The Dynamic ECME Algorithm

Yunxiao He

Yale University, New Haven, USA

Chuanhai Liu

Purdue University, West Lafayette, USA

Summary. The Expectation/Conditional Maximisation Either (ECME) algorithm has proven to be an effective way of accelerating the Expectation Maximisation (EM) algorithm for many problems. Recognising the limitation of using prefixed acceleration subspaces in ECME, we propose a Dynamic ECME (DECME) algorithm which allows the acceleration subspaces to be chosen dynamically. The simplest DECME implementation is what we call DECME-1, which uses the line determined by the two most recent estimates as the acceleration subspace. The investigation of DECME-1 leads to an efficient, simple, stable, and widely applicable DECME implementation, which uses two-dimensional acceleration subspaces and is referred to as DECME-2s. The fast convergence of DECME-2s is established by the theoretical result that in a small neighbourhood of the maximum likelihood estimate (MLE), it is equivalent to a conjugate direction method. The remarkable accelerating effect of DECME-2s and its variant is also demonstrated with multiple numerical examples.

Keywords: Conjugate direction; EM algorithm; ECM algorithm; ECME algorithm; Square iterative methods; Successive overrelaxation.

1. Introduction

After its booming popularity of more than 30 years since the publication of Dempster et al. (1977), the EM algorithm is still expanding its application scope in various areas. At the same time, to overcome the slow convergence of EM, quite a few extensions of EM have been developed in such a way that they run faster than EM while maintaining its attractive simplicity and stability. We refer to Varadhan and Roland (2008) for a recent nice review of various methods for accelerating EM. In the present paper, we start by exploring the convergence of the ECME algorithm (Liu and Rubin, 1994), which has proved to be a simple and effective method to accelerate its parent EM algorithm; see, *e.g.*, Sammel and Ryan (1996), Kowalski et al. (1997), and Pinheiro et al. (2001), to name a few.

ECME is a simple extension of the Expectation/Conditional Maximisation (ECM) algorithm (Meng and Rubin, 1993) which itself is an extension of EM. These three algorithms are summarised as follows. Let Y_{obs} be the observed data. Denote by $L(\theta|Y_{obs}),\ \ \theta \in \Theta \subset \mathcal{R}^p$, the observed log-likelihood function of θ. The problem is to find the MLE $\hat{\theta}$ that maximises $L(\theta|Y_{obs})$. Let $Y = (Y_{obs},\ Y_{mis})$ represent the complete data with Y_{obs} augmented by the missing data Y_{mis}. As an iterative algorithm, the t-th iteration of EM consists of an E-step, which computes $Q(\theta|Y_{obs}, \theta_{t-1})$, the expected complete-data log-likelihood function given

the observed data and the current estimate θ_{t-1} of θ, and an M-step, which finds $\theta = \theta_t$ to maximise $Q(\theta|Y_{obs}, \theta_{t-1})$.

The ECM algorithm replaces the M-step of EM with a sequence of simpler constrained or conditional maximisation (CM) steps, indexed by $s = 1, \cdots, S$, each of which fixes some function of θ, $h_s(\theta)$. The ECME algorithm further partitions the S CM-steps into two groups $\mathscr{S}_Q$ and $\mathscr{S}_L$ with $\mathscr{S}_Q \cup \mathscr{S}_L = \{1, \cdots, S\}$. While the CM-steps indexed by $s \in \mathscr{S}_Q$ (refereed to as the MQ-steps) remain the same as with ECM, the CM-steps indexed by $s \in \mathscr{S}_L$ (refereed to as the ML-steps) maximise $L(\theta|Y_{obs})$ in the subspace induced by $h_s(\theta)$. A more general framework that includes ECM and ECME as special cases is developed in Meng and van Dyk (1997). However, most of the practical algorithms developed under this umbrella belong to the scope of a simple case, *i.e.,* the parameter constraints are formed by creating a partition, $\mathscr{P}$, of θ as $(\theta_1, \cdots, \theta_S)$ with associated dimensions $(d_1, \cdots, d_S)$. Mathematically we have $h_s(\theta) = (\theta_1, \cdots, \theta_{s-1}, \theta_{s+1}, \cdots, \theta_S)$ for $s = 1, \cdots, S$.

The advantage of ECME over EM in terms of efficiency depends on the relationship between the slowest converging directions of EM and the acceleration subspaces of ECME, *i.e.*, the subspaces for the ML-steps. For example, when the former is effectively embedded within the latter, ECME achieves its superior gain of efficiency over its parent EM. In practise, we usually have no information about the convergence of EM before obtaining the MLE and cannot select the prefixed acceleration subspaces of ECME accordingly. Hence small or minor efficiency gain by ECME is expected in some situations. This is illustrated by the two examples in Section 2 and motivates the idea of dynamically constructing subspaces for applying the ML-step. This idea is formulated as the generic DECME algorithm. The simplest implementation of DECME is what we call DECME-1. It uses the line determined by the two most recent estimates as the acceleration subspace for the next ML-step and is similar to the classical Successive Overrelaxation (SOR) method (Frankel, 1950; Young, 1954; Salakhutdinov and Roweis, 2003; Hesterberg, 2005, among many others although sometimes under different names). However, as shown later in Section 3.2, DECME-1 suffers from what is known as the zigzagging problem (see Figure 1 for an illustration).

Motivated by the zigzagging phenomenon observed on DECME-1, we propose an efficient DECME implementation, called DECME-2. It is shown that, under some common assumptions, DECME-2 is equivalent to a conjugate direction method, which has been proposed in several different contexts, *e.g.,* solving linear systems (Concus et al., 1976) and nonorthogonal analysis of variance (Golub and Nash, 1982). Jamshidian and Jennrich (1993) propose to use the conjugate direction method to accelerate EM. They call the resulting method AEM and demonstrate its dramatically improved efficiency. However, AEM is not as popular as one would expect it to be. This is perhaps due to its demands for extra efforts for coding the gradient vector of $L(\theta|Y_{obs})$, which is problem specific and can be expensive to evaluate.

Compared to AEM, DECME-2 is simpler to implement because it does not require computing the gradient of $L(\theta|Y_{obs})$. As DECME-1, the only extra requirement for implementing DECME-2 is a line search scheme which can be used for almost all EM algorithms. Furthermore, the required loglikelihood evaluation is typically coded with EM implementation for debugging and monitoring convergence. However, it is known that the line search

schemes without access to gradient information can take a large number of loglikelihood evaluations to run. To remedy this problem, we propose a simplified version of DECME-2, called DECME-2s . DECME-2s can significantly reduce the number of loglikelihood evaluations required in each iteration while maintaining the fast convergence rate of DECME-2. Numerical results show that DECME-2 and DECME-2s achieve dramatic improvement over EM in terms of both the number of iterations and CPU time.

The remaining of the paper is arranged as follows. Section 2 provides a pair of motivating ECME examples. Section 3 defines the generic DECME algorithm, discusses the convergence of DECME-1, and proposes the two efficient novel implementations of DECME. The relation between the DECME algorithm and the SQUAREM algorithm of Varadhan and Roland (2008) is also discussed. Section 4 presents several numerical examples to compare the performance of the two new methods and other state-of-art EM accelerators. Section 5 concludes with a few remarks.

2. Two Motivating ECME Examples

Following Dempster et al. (1977), in a small neighbourhood of $\hat{\theta}$, we have approximately

$$\hat{\theta} - \theta_t = DM^{EM}(\hat{\theta} - \theta_{t-1}), \tag{1}$$

where the $p \times p$ matrix DM^{EM} is known as the missing information fraction and determines the convergence rate of EM. More specifically, each eigenvalue of DM^{EM} determines the convergence rate of EM along the direction of its corresponding eigenvector (see Appendix A).

It is shown in Liu and Rubin (1994) that ECME also has a linear convergence rate determined by the $p \times p$ matrix DM^{ECME} that plays the same role for ECME as DM^{EM} does for EM. Obviously, ECME will be faster than EM if the largest eigenvalue of DM^{ECME} is smaller than that of DM^{EM}. With the following two examples we illustrate that it is the choice of the acceleration subspaces by ECME that determines the relative magnitude of the dominating eigenvalues of DM^{EM} and DM^{ECME}, and hence the relative efficiency of EM and ECME. All the numerical examples in this paper are implemented in R (R Development Core Team, 2010).

2.1. A Linear Mixed-effects Example

Consider the rat population growth data in Gelfand et al. (1990, Tables 3, 4). Sixty young rats were assigned to a control group and a treatment group with $n = 30$ rats in each. The weight of each rat was measured at ages $x = 8,\ 15,\ 22,\ 29,$ and 36 days. We denote by y_i^g the weights of the ith rat in group g with $g = c$ for the control group and $g = t$ for the treatment group. The following linear mixed-effects model (Laird and Ware, 1982) is considered in Liu (1998):

$$y_i^g|\theta \sim N(X\beta_g + Xb_i^g,\ \sigma_g^2 I_5), \quad b_i^g \sim N(0, \Psi), \tag{2}$$

for $i = 1, \cdots, n$ and $g = c$ and t, where X is the 5×2 design matrix with a vector of ones as its first column and the vector of the five age-points as its second column, $\beta_g = (\beta_{g,1}, \beta_{g,2})'$

contains the fixed effects, $b_i^g = (b_{i,1}^{(g)}, b_{i,2}^{(g)})'$ contains the random effects, $\Psi > 0$ is the 2×2 covariance matrix of the random effects, and θ denotes the vector of the parameters, i.e., $\theta = (\beta_{c,1}, \beta_{c,2}, \beta_{t,1}, \beta_{t,2}, \Psi_{1,1}, \Psi_{1,2}, \Psi_{2,2}, \sigma_c^2, \sigma_t^2)'$. Let $\beta = (\beta_{c,1}, \beta_{c,2}, \beta_{t,1}, \beta_{t,2})'$ and $\sigma^2 = (\sigma_c^2, \sigma_t^2)'$. The starting point for running EM and ECME is chosen to be $\beta = (0, 0, 0, 0)'$, $\sigma^2 = (1, 1)'$, and $\Psi = I_2$. The stopping criterion $C1$ in (8) was used (see Section 4.1).

For this example, ECME converges dramatically faster than EM. Specifically, it takes $5,968$ iterations for EM to converge. With the same setting, ECME (version 1 in Liu and Rubin (1994) with $\theta_{\mathscr{P}_Q} = (\Psi_{1,1}, \Psi_{1,2}, \Psi_{2,2}, \sigma^{2'})'$ and $\theta_{\mathscr{P}_L} = \beta$) uses only 20 iterations. The gain of ECME over EM is explained clearly by the relation between the slow converging directions of EM and the partition of the parameter space for ECME. From Table 1, the two largest eigenvalues of DM^{EM} are 0.9860 and 0.9746, which are close to 1 and, thereby, make EM converge very slowly. From Table 2, it is clear that the first four "worst" directions of EM fall entirely in the subspace determined by the fixed effect β. Since $\theta_{\mathscr{S}_L} = \beta$ for ECME, the slow convergence of EM induced by the four slowest directions is diminished by implementing the ML-step along the subspace of β. This is clear from the ECME row in Table 1, where we see the four largest eigenvalues of DM^{EM} become 0 in DM^{ECME} while the five small eigenvalues of DM^{EM} remain the same for DM^{ECME}.

2.2. A Factor Analysis Example

Consider the confirmatory factor analysis model example in Jöreskog (1969), Rubin and Thayer (1982), and Liu and Rubin (1998). The data is provided in Liu and Rubin (1998) and the model is as follows. Let Y be the observable nine-dimensional variable on an unobservable variable Z consisting of four factors. For n independent observations of Y, we have

$$Y_i|(Z_i, \beta, \sigma^2) \sim N(Z_i\beta, \mathsf{diag}(\sigma_1^2, \cdots, \sigma_9^2)), \tag{3}$$

where β is the 4×9 factor-loading matrix, $\sigma^2 = (\sigma_1^2, \cdots, \sigma_9^2)'$ is called the vector of uniquenesses, and given (β, σ^2), $Z_1, \cdots, Z_n$ are independently and identically distributed with $Z_i \sim N(0, I_4)$ for $i = 1, \cdots, n$. In the model, there are zero factor loadings on both factor 4 for variables 1-4 and on factor 3 for variables 5-9. Let $\beta_{j\cdot}$, $j = 1, \cdots, 4$, be the four rows of β, then the vector of the 36 free parameters is $\theta = (\beta_{1\cdot}, \beta_{2\cdot}, \beta_{3,1-4}, \beta_{4,5-9}, \sigma^2)'$. Liu and Rubin (1998) provided detailed comparison between EM and ECME, which shows that the gain of ECME over EM is impressive, but not as significant as ECME for the previous linear mixed-effects model example in Section 2.1.

The slow convergence of EM for this example is easy to explain from Table 3 which shows that DM^{EM} has multiple eigenvalues close to 1. From Table 4, the eigenvector corresponding to the dominant eigenvalue of DM^{EM} falls entirely in the subspace spanned by β_1 and β_2. This clearly adds difficulty to the ECME version used by Liu and Rubin (1998) where $\theta_{\mathscr{S}_Q} = (\beta_{1\cdot}, \beta_{2\cdot}, \beta_{3,1-4}, \beta_{4,5-9})'$ and $\theta_{\mathscr{S}_L} = \sigma^2$. For this version of ECME, the eigenvalues of DM^{ECME} are given in the ECME-1 row of Table 3, where we see that the dominant eigenvalue of DM^{EM} remains unchanged for DM^{ECME}. To eliminate the effect of the slowest direction of EM, we can try another version of ECME by letting $\theta_{\mathscr{S}_Q} = \sigma^2$

and $\theta_{\mathscr{S}_L} = (\beta_{1\cdot}, \beta_{2\cdot}, \beta_{3,1-4}, \beta_{4,5-9})'$. The eigenvalues of DM^{ECME} for this version are given in the ECME-2 row of Table 3. Although the second version of ECME is more efficient than the first version, it is difficult in general to eliminate all the large eigenvalues in DM^{EM} by accelerating EM in a fixed subspace. For example, the eigenvector corresponding to the second largest eigenvalue of DM^{EM} shown in Table 4 is not in the subspace spanned by any subset of the parameters.

3. The DECME Algorithm

3.1. The Generic DECME Algorithm

As shown in last section, the efficiency gain of ECME over EM based on static choices of the acceleration subspaces may be limited since the slowest converging directions of EM depend on both the data and the model. It is thus expected to have a great potential to construct the acceleration subspaces dynamically based on, for example, the information from past iterations. This idea is formulated as the following generic DECME algorithm. At the t^{th} iteration of DECME, the algorithm proceeds as follows.

> **The Generic DECME Algorithm: the t^{th} iteration with the input $\tilde{\theta}_{t-1}$**
> **E-step:** This is the same as the E-step of the original EM algorithm;
> **M-step:** Run the following two steps:
>
> > **CM-step:** Compute $\theta_t = \mathsf{argmax}_\theta Q(\theta|\tilde{\theta}_{t-1})$ as in the original EM algorithm;
> >
> > **Dynamic CM-step:** Compute $\tilde{\theta}_t = \mathsf{argmax}_{\theta\in\mathscr{V}_t} L(\theta|Y_{obs})$, where $\mathscr{V}_t$ is a low-dimensional subspace with $\theta_t \in \mathscr{V}_t$.

As noted in Meng and van Dyk (1997), the ML-steps in ECME should be carried out after the MQ-steps to ensure convergence. Under this condition, ECME with only a single ML-step is obviously a special case of DECME. In cases where multiple ML-steps are performed in ECME, a slightly relaxed version of the Dynamic CM-step, *i.e.,* simply computing $\tilde{\theta}_t$ such that $L(\tilde{\theta}_t|Y_{obs}) \geq L(\theta_t|Y_{obs})$, will still make DECME a generalisation of ECME. In either case, the monotone increase of the loglikelihood function in DECME is guaranteed by the nested EM algorithm (Dempster et al., 1977; Wu, 1983), which ensures the stability of DECME. The convergence rate of DECME relies on the structure of the specific implementation, *i.e.,* how $\mathscr{V}_t$ is constructed.

3.2. DECME-1: a Simple but Inefficient Special Case of DECME

Let $\{\theta_t - \tilde{\theta}_{t-1}\}$ represent the linear subspace spanned by the vector $\theta_t - \tilde{\theta}_{t-1}$. DECME-1 is obtained by specifying $\mathscr{V}_t = \theta_t + \{\theta_t - \tilde{\theta}_{t-1}\}$ in the Dynamic CM-step of DECME, *i.e.*, $\tilde{\theta}_t = \theta_t + \alpha_t d_t$, $d_t = \theta_t - \tilde{\theta}_{t-1}$, and $\alpha_t = \mathsf{argmax}_\alpha L(\theta_t + \alpha d_t|Y_{obs})$. The so-called relaxation factor α_t can be obtained by a line search. See Figure 2 for an illustration of the DECME-1 iteration.

The reason that DECME-1 can be used to accelerate EM is clear from the following theorem with the proof given in Appendix B, which implies that in a small neighbourhood

of the MLE, a point with larger loglikelihood value can always be found by enlarging the step size of EM.

THEOREM 3.1. *In a small neighbourhood of $\hat{\theta}$ (or equivalently, for sufficiently large t), the relaxation factor α_t of DECME-1 is always positive.*

The conservative EM updates are illustrated in Figure 1 for a two-dimensional artificial example. For simplicity, it has also been proposed to choose α_t as a fixed positive number (*e.g.,* Lange, 1995). We call this version with fixed α_t the SORF method. Let λ_1 and λ_p be the largest and smallest eigenvalues of $I_{com}^{-1} I_{obs}$ (see Appendix B for detailed discussion). It is well known that SORF achieves its optimal convergence rate $(\lambda_1 - \lambda_p)/(\lambda_1 + \lambda_p)$ if $\alpha_t = 2/(\lambda_1 + \lambda_p) - 1$ for any t (see, *e.g.*, Salakhutdinov and Roweis, 2003; Roland, 2010). In the past, the theoretical argument for DECME-1 has been mainly based on this fact, which is obviously insufficient. The following theorem with the proof given in Appendix C sheds new lights on the convergence of DECME-1.

THEOREM 3.2. *For a two-dimensional problem (i.e., $p = 2$) and in a small neighbourhood of $\hat{\theta}$ (or equivalently, for sufficiently large t), the following results hold for SOR:*

1.) $\alpha_t = \alpha_{t-2}$;

2.) DECME-1 converges at least as fast as the optimal SORF, and the optimal SORF converges faster than EM; and

3.) DECME-1 oscillates around the slowest converging direction of EM; The DECME-1 estimates from the odd-numbered iterations lie on the same line and so do those from the even-numbered iterations; Furthermore, the two lines intersect at the MLE $\hat{\theta}$.

The zigzagging phenomena of DECME-1 revealed by conclusion 3 is illustrated in Figure 1. For the case of $p > 2$, it is interesting to see that the relaxation factors α_t generated from DECME-1 also have a similar oscillating pattern as that for $p = 2$ (conclusion 1). This is illustrated in Figure 3. The top panel of Figure 3 shows the relaxation factors for the two-dimensional example used to generate Figure 1 and the lower panel shows those for a nine-dimensional simulated example. The nine-dimensional example is generated by simulating the behaviour of EM in a small neighbourhood of the MLE for the linear mixed-effects example in Section 2.1.

3.3. DECME-2: a Basic Scheme with Quadratic Convergence

The zigzagging phenomenon of DECME-1 shown in Figure 1 suggests intuitively a line search along the line connecting the zigzag points, as shown by one of the red dashed lines. The resulting procedure converges immediately for two-dimensional quadratic functions, basic approximations for iterative optimization methods. This motivated us to consider a basic scheme that effectively considers two-dimensional acceleration subspaces. Such a basic scheme, called DECME-2, is discussed here. It is shown that DECME-2 has a very attractive theoretical property, i.e., it is equivalent to a conjugate direction method; see Theorem 3.3.

Given the starting point $\tilde{\theta}_0$, DECME-2 takes one DECME-1 iteration to obtain $\tilde{\theta}_1$. The t^{th} $(t > 1)$ iteration of DECME-2 consists of two steps: a DECME-1 iteration producing an intermediate point $\tilde{\theta}_t^\star$ and a line search along the line connecting $\tilde{\theta}_{t-2}$ and $\tilde{\theta}_t^\star$ to obtain $\tilde{\theta}_t$. This algorithm is illustrated in Figure 2 is formally described in Table 5 in the framework of the generic DECME algorithm. We note that $\tilde{\theta}_t$ is actually the point that maximises $L(\theta|Y_{obs})$ over the two-dimensional subspace $\mathscr{V}_t = \tilde{\theta}_{t-1} + \{\tilde{\theta}_{t-1} - \tilde{\theta}_{t-2},\ \theta_t - \tilde{\theta}_{t-1}\}$ under certain conditions. This can be seen from the proof, given in Appendix D, of the following theorem, which demonstrates the efficiency of DECME-2.

THEOREM 3.3. *In a small neighbourhood of the MLE, DECME-2 with exact line search is equivalent to the conjugate direction method AEM.*

The most notable property of a conjugate direction method is that it converges to MLE after at most p iterations, assuming that the loglikelihood is a quadratic function of p parameters (Luenberger, 2003, *pg.* 240). In light of this observation, it is a common practise to restart a conjugate direction method after a complete cycle of p iterations for nonquadratic problems. More importantly, this property shows that a complete cycle of a conjugate direction method solves a quadratic problem exactly as Newton-Raphson's method does in one iteration. It can be proved that a conjugate direction method like DECME-2 has a convergence rate of order two *w.r.t.* each complete cycle of p iterations (Luenberger, 2003, *pg.* 254). This implies a dramatic improvement of DECME-2 over EM.

As noted in Section 1, the evaluation of the gradient vector of $L(\theta|Y_{\text{obs}})$ required for AEM is problem specific and thus demands substantial programming efforts. Compared to AEM, DECME-2 is much easier to implement since the line search scheme can be performed by utilizing the line search functions provided in popular software packages. For example, the *optimise* function in R was used for our numerical examples in Section 4. In general, these line search functions are developed primarily for unconstrained problems. Our way of using the *optimise* function is described as follows. Denote the current estimate by θ and the search direction by d. The feasible region, *i.e.,* an interval $\mathcal{I}$ such that $\theta + \alpha\mathsf{d}$ is valid for any $\alpha \in \mathcal{I}$, is first computed. Then the line search is strictly enforced inside the feasible region by passing the interval $\mathcal{I}$ to the *optimise* function through its option *interval=*; see detailed discussion about the computation of $\mathcal{I}$ in Appendix E. Furthermore, the accuracy of the line search can be controlled, *e.g.*, by setting $tol = 0.01$ in the *optimise* function.

The idea behind DECME-2 is very similar to the parallel tangent (PARTAN) method for accelerating the steepest descent method (Shah et al., 1964). PARTAN can be viewed as a particular implementation of the conjugate gradient method (Fletcher and Reeves, 1964), built upon the method of Hestenes and Stiefel (1952) for solving linear systems. It is also worth noting that PARTAN has certain advantages over the conjugate gradient method as discussed in Luenberger (2003, p. 257). For example, the convergence of PARTAN is more reliable than the conjugate gradient method when inexact line search is used as is often the case in practise.

3.4. DECME-2s : an Efficient DECME Implementation

While DECME-2 has the attractive quadratic convergence, its perfornamce in terms of CPU can be further improved by reducing the CPU time for the two required line searches. Early such efforts can be found in He (2009). A new version, called DECME-2s, is obtained by replacing the two (inexact) line searches of DECME-2 with a single inexact two-dimensional search based on a two-dimenional quadratic approximation. Although the idea is simple, as with one-dimensional or line search, implementation of a two-dimensional or plane search is somewhat tedious and is given below. However, it should be noted that the two line searches in DECME-2 can be replaced with any method optimising the two-dimensional function $f_t(x,\ y) = L[\theta_t + x(\theta_t - \tilde{\theta}_{t-1}) + y(\theta_t - \tilde{\theta}_{t-2})|Y_{obs}]$ without compromising the convergence rate.

Denote by $f_t(x,\ y) = L[\theta_t + x(\theta_t - \tilde{\theta}_{t-1}) + y(\theta_t - \tilde{\theta}_{t-2})|Y_{obs}]$ the two-dimensional objective function, i.e., the actual likelihood function restricted over the two-dimensional acceleration subspace of DECME-2. We construct and optimise the quadratic approximation to $f_t(x,\ y)$:

$$f_t^*(x,\ y) = f_0 + (x, y)(a,\ b)' + (x, y)H(x,\ y)', \tag{4}$$

which has six free parameters f_0, a, b, and two diagonal element (c, d) and an off-diagonal element e of the 2×2 symmetric matrix H.

To determine the function $f_t^*(x,\ y)$, six points with their corresponding loglikelihood values are required. Three pairs, $(\theta_t,\ L(\theta_t|Y_{obs}))$, $(\tilde{\theta}_{t-1},\ L(\tilde{\theta}_{t-1}|Y_{obs}))$, and $(\tilde{\theta}_{t-2},\ L(\tilde{\theta}_{t-2}|Y_{obs}))$, are already avaialble. The other three points, denoted by $\xi_t^{(i)}$ $(i = 1, 2, 3)$, are chosen from the following three directions: $\mathsf{d}_t^{(1)} = \theta_t - \tilde{\theta}_{t-1}$, $\mathsf{d}_t^{(2)} = \theta_t - \tilde{\theta}_{t-2}$, and $\mathsf{d}_t^{(3)} = \tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}$.

Although not explicitly stated, we have utilized the conventional coordinate system with the origin θ_t and the basis $(\theta_t - \tilde{\theta}_{t-1},\ \theta_t - \tilde{\theta}_{t-2})$ in defining the two functions $f_t(x,\ y)$ and $f_t^*(x,\ y)$. Under this convenient coordinate system, the three points θ_t, $\tilde{\theta}_{t-1}$, and $\tilde{\theta}_{t-2}$ have coordinates $(0, 0)$, $(-1, 0)$, and $(0, -1)$, respectively. For each direction $\mathsf{d}_t^{(i)}$, let $\bar{\alpha}_t^{(i)}$ be the maximum feasible step size such that $\theta_t + \alpha_t^{(i)}\mathsf{d}_t^{(i)}$ is a valid estimate of θ for any $\alpha_t^{(i)} \in [0, \bar{\alpha}_t^{(i)})$ (see appendix E for more discussion). Then $\xi_t^{(i)} = \theta_t + \alpha_t^{(i)} d_t^{(i)}$, where $\alpha_t^{(i)} = \mathsf{min}(1,\ 0.9\bar{\alpha}_t^{(i)})$. When $\alpha_t^{(1)} = 1$, $\xi_t^{(1)}$ and $\tilde{\theta}_{t-1}$ are symmetric about θ_t. The same is true for $\xi_t^{(2)}$ and $\tilde{\theta}_{t-2}$ if $\alpha_t^{(2)} = 1$. The particular number 0.9 is used to prevent $\xi_t^{(i)}$ from being too close to the boundary. In the above conventional coordinate system, the coordinates of $\xi_t^{(i)}$, $i = 1, 2, 3$, are $(\alpha_t^{(1)}, 0)$, $(0, \alpha_t^{(2)})$, and $(\alpha_t^{(3)}, -\alpha_t^{(3)})$, respectively.

The stationary point $(\hat{x}, \hat{y})$ of $f_t^*(x, y)$ can then be easily computed as

$$(\hat{x}_t, \hat{y}_t) = \frac{1}{2}(a,\ b)H^{-1}, \tag{5}$$

where, suppressing Y_{obs} for readability,

$$a = \frac{L(\xi_t^{(1)}) - L(\theta_t) - (\alpha_t^{(1)})^2[L(\tilde{\theta}_{t-1}) - L(\theta_t)]}{\alpha_t^{(1)} + (\alpha_t^{(1)})^2},\ b = \frac{L(\xi_t^{(2)}) - L(\theta_t) - (\alpha_t^{(2)})^2[L(\tilde{\theta}_{t-2}) - L(\theta_t)]}{\alpha_t^{(2)} + (\alpha_t^{(2)})^2}, \tag{6}$$

and the elements of H are determined by

$$\begin{aligned} &c = L(\tilde{\theta}_{t-1}) - L(\theta_t) + a,\ d = L(\tilde{\theta}_{t-2}) - L(\theta_t) + b, \\ &e = -\frac{L(\xi_t^{(3)}) - L(\theta_t) - (a-b)\alpha_t^{(3)} - (c+d)(\alpha_t^{(3)})^2}{2(\alpha_t^{(3)})^2}. \end{aligned} \tag{7}$$

Let $\mathsf{d}_t^{(4)} = \hat{x}_t(\theta_t - \tilde{\theta}_{t-1}) + \hat{y}_t(\theta_t - \tilde{\theta}_{t-2})$ and similarly find the maximum feasible step size $\bar{\alpha}_t^{(4)}$. Compute $\xi_t^{(4)} = \theta_t + \alpha_t^{(4)}\mathsf{d}_t^{(4)}$, where $\alpha_t^{(4)} = \mathsf{min}(1, 0.9\bar{\alpha}_t^{(4)})$. Then $\tilde{\theta}_t$ is chosen to be the point with maximum loglikelihood values among $\{\theta_t,\ \xi_t^{(1)},\ \xi_t^{(2)},\ \xi_t^{(3)},\ \xi_t^{(4)}\}$.

Although individual $\xi_t^{(i)}$'s, $i = 1, \cdots, 4$, are not necessarily better than θ_t in terms of loglikelihood, it is easy to see that $L(\tilde{\theta}_t|Y_{obs}) \geq L(\theta_t|Y_{obs})$. In addition, $(\hat{x}_t,\ \hat{y}_t)$ maximises the function $f_t^*(x,\ y)$ when $-H$ is positive definite. Thus the point $\xi_t^{(4)}$ is expected to be very close to the maximum of $f_t(x, y)$, especially when the current estimate is near the MLE.

A one-dimensional quadratic function can be similarly constructed for computing the first and the restarting iterations. However, to keep the programming simple, we didn't perform the restarting step for DECME-2s in our numerical examples. In the first iteration of DECME-2s, only one EM iteration is computed without doing any acceleration.

An illustration of DECME-2s is provided in Figure 2. A pseudocode summarising the computations in DECME-2s is provided in Table 6. The first iteration of DECME-2s requires one EM iteration and two loglikelihood evaluations. For all other DECME-2s iterations, one EM iteration and five loglikelihood evaluations are required. When terminated after t iterations, DECME-2s uses t EM iterations and $5t - 3$ loglikelihood evaluations, resulting in a dramatic reduction from DECME-2.

3.5. Relationship to the SQUAREM Algorithm

DECME-2 is not the first algorithm that is motivated by the slow convergence of EM and SOR-like algorithms such as DECME-1. In Varadhan and Roland (2008), an analogy was made between SOR (under the name of Steffensen-type methods for EM or simply STEM) and the gradient descent method (under the name of Cauchy's method). An idea for accelerating the gradient descent (the Cauchy-Barzilai-Borwein method) was then adopted to accelerate EM and the resulting algorithm was called the squared iterative method (SQUAREM). Several versions of SQUAREM were proposed and the scheme SqS3 was recommended in Varadhan and Roland (2008). Two different implementations of SqS3, SQUAREM1 and SQUAREM2, have been developed by the authors of Varadhan and Roland (2008), where loglikelihood evaluation is required by SQUAREM1, but not by SQUAREM2. Empirical results in Varadhan and Roland (2008) demonstrated the superiority of SQUAREM over EM and SOR or DECME-1.

There are some similarities between DECME and SQUAREM. Both DECME-2 and SQUAREM use two-dimensional acceleration subspaces. However, they are quite different in many aspects. Each iteration of DECME-2 and DECME-2s contains one EM iteration. A two-dimensional acceleration subspace is then determined by the current EM iteration and the previous DECME iteration. SQUAREM requires two EM iterations for the "square" acceleration step (step 7 in Table 1 of Varadhan and Roland, 2008). Furthermore, Theorem 3.3 provides a strong theoretical adjustment for the fast convergence of DECME-2 and DECME-2s .

Another major distinction between DECME and SQUAREM comes from the different means of stabilizing the algorithms. A common problem for acceleration schemes such as SQUAREM and DECME is that the model constraints cannot be automatically handled.

The details of our treatment of this problem are given in Sections 3.3, 3.4, and Appendix E. Although these procedures are very simple to implement, they do need some problem-specific information. In SQUAREM, the third EM iteration (step 8 in Table 1 of Varadhan and Roland, 2008) is not required for acceleration purpose. Rather, it is utilized primarily for stabilizing the algorithm. In addition, some other stabilization techniques are also used in SQUAREM1 and SQUAREM2. For example, in SQUAREM1 the loglikelihood is used for detecting violations to constraints: an estimate is considered to be invalid if the loglikelihood cannot be computed. The application of these stabilizing strategies in SQUAREM avoids using any problem-specific information except for the EM iteration itself.

It is easy to see that the stabilization methods in SQUAREM can also be used in DECME to avoid the simple model-specific procedures for feasible region calculation. However, we didn't adopt their methods in our implementation since these strategies may fail for some commonly used models. For many statistical models, the model constraints are sometimes more restrictive than the constraints defining the domain of the loglikelihood. For instance, in the factor analysis example in Section 2.2, the loglikelihood computation merely requires that the variance matrix $\beta'\beta + \mathsf{diag}(\sigma^2)$ to be positive definite. However, the factor analysis model specifies that $\mathsf{diag}(\sigma^2)$ is positive definite, which defines a feasible region smaller than the domain of the loglikelihood. Similar problems also arise for the mixed-effects example in Section 2.1 and the multivariate Student-t example in Section 4.4. As a consequence, the model constraints can still be violated even loglikelihood is used as a safeguard. These points are further illustrated with the numerical examples in Section 4.

4. Numerical Examples

One of the advantages of DECME and SQUAREM is that they can be applied whenever EM is available. Two other popular EM accelerators, the Alternating ECM (AECM) algorithm of Meng and van Dyk (1997) and the parameter-expanded EM (PX-EM) algorithm of Liu et al. (1998), accelerate EM by making use of specific model structures. As a consequence, they can only be used in limited situations. In this section, we use four examples to compare the performance of EM, DECME-2, DECME-2s , SQUAREM1, and SQUAREM2. The comparison to PX-EM and AECM is done for two of the four examples for which they are available. We omit the comparison with DECME-1 here because it is conclusive that DECME-1 is less favorable based on the theoretical results in Section 3.2 and the simulation results in previous studies.

4.1. The Setting for the Numerical Experiments

Our comparison study focuses on the efficiency and stability of different algorithms. Here we discuss our choice of starting values, stopping criteria, and the different measures we report for all examples except for the normal-mixture example in Section 4.5, where slightly different criteria were used for the simulation study involvs many simulated data sets.

All algorithms were run $5,000$ times with randomly generated starting points. Each run was terminated by a stopping rule combining three criteria (with minor modifications for

SQUAREM, see below):

$$\begin{aligned} &C1 : L(\tilde{\theta}_t|Y_{obs}) - L_{max} > -\epsilon_L; \\ &C2 : \|\tilde{\theta}_t - \tilde{\theta}_{t-1}\|_2 < \epsilon_\theta; \\ &C3 : t > t_{max}. \end{aligned} \tag{8}$$

The maximum log-likelihood L_{max} used in C1 was obtained by running EM with reasonable starting values and very stringent stopping criteria. Criterion $C2$ is commonly used in practise. However, the use of $C2$ alone for comparing different methods is unfair because fast algorithms usually converge to better estimates (in terms of loglikelihood) than slow algorithms. Criterion $C1$ was used to resolve this problem. In addition, $C1$ also prevents unnecessary runs for slow algorithms since it can take a large number of iterations for them to fine-tune the estimation. A large value of t_{max} in C3 was used to allow the algorithms to run sufficiently long enough to explore their convergence behavior. For SQUAREM, we kept the two stopping criteria used in the code developed by its authors ($C3$ and a modification of $C2$); their code was only modified to enforce $C1$.

We set $\epsilon_L = 10^{-6}$, $\epsilon_\theta = 10^{-8}$, and $t_{max} = 30,000$. Under this configuration, $C2$ and $C3$ are usually more stringent than $C1$. As a consequence, $C2$ and $C3$ only function if an algorithm moves very slowly around a non-optimal estimation or if the algorithm converges to a local rather than the global maximum. If an algorithm was terminated by $C2$ or $C3$, the gradient of the loglikelihood at the final estimation was calculated using the *grad* function in the R package *numDeriv*. This is used to verify if the final estimate is a maximum. Denote the calculated gradient by $\tilde{g}$ and let $\|\tilde{g}\|_\infty$ be its l_∞ norm, the largest absolute component of $\tilde{g}$. To report the numerical results, each run of each algorithm was classified into one of the three categories:

1) "successful": either the program was terminated by $C1$ or $\|\tilde{g}\|_\infty < \epsilon_g$, $\epsilon_g = 10^{-4}$;

2) "non-optimal": the program was terminated by $C2$ or $C3$, but $\|\tilde{g}\|_\infty \geq \epsilon_g$;

3) "invalid": either the program crashed without providing any estimate or it produced an invalid estimate, *i.e.,* (i) the model constraints were violated, or (ii) the estimate was so close to the boundary that the gradient was not computable by the R function *grad*.

The results for the first three examples are summarised in Tables 7-11. For each algorithm, we report the number of "successful" cases (#(suc)), "non-optimal" cases (#(non)), and "invalid" cases (#(inv)) based on the 5,000 runs. These results facilitate the comparison of stability. As discussed in Varadhan and Roland (2008), the number of EM iterations measures the rate of convergence, while the CPU time is a measure of the overall computational efficiency. For comparison in terms of efficiency, we report the mean and the standard deviation (in bracket) of the number of EM iterations (#(EM)), number of loglikelihood evaluations (#(llk)), and CPU time (CPU (s)) from the "successful" runs. To make the comparison more meaningful, the mean and the standard deviation were computed after excluding extreme values with the following simple rule of thumb: the observations beyond

$3 * IQR$ are considered as extreme, where IQR represents the interquartile range. The number of observations/runs after excluding extreme values (#(obs)) is also reported.

It should be noted that EM, PX-EM, AECM, and SQUAREM2 do not need loglikelihood evaluations originally. To implement criterion $C1$, we computed the loglikelihood once in each iteration, which enforced some seemingly unnecessary overhead. However, it is a common practise to monitor the loglikelihood increase when implementing EM-type algorithms. Moreover, one loglikelihood evaluation is typically much faster than one EM iteration. Hence, this treatment does not change the overall conclusion of the comparison. This is especially true for the comparison to SQUAREM because each iteration of SQUAREM runs three EM iterations with only one loglikelihood evaluation.

4.2. The Linear Mixed-effects Example

Consider the mixed-effects example of Section 2.1. Since the convergence speed of EM was very sensitive to the choice of starting values for this example, we used two different sets of starting values. The first set has the starting values with (i) each component of the unconstrained β chosen from $U(-10,\ 10)$, where $U(a,\ b)$ represents the uniform random distribution over the interval (a, b), (ii) each of $\Psi_{1,1}$, $\Psi_{2,2}$, σ_c^2, and σ_t^2 chosen from $U(0.1,\ 20)$, and (iii) $\Psi_{1,2}$ set to 0. The second set was obtained similarly, but with the two uniform distributions replaced by $U(-50,\ 50)$ and $U(0.1,\ 100)$. Since EM can be extremely slow for the second set of starting values, we set $t_{max} = 2 \times 10^5$ for criterion $C3$ in (8). The corresponding results are shown in Tables 7 and 8.

The columns #(suc/non/inv) in Tables 7 and 8 show that EM almost never failed for this example. While EM is obviously the best in terms of stability, SQUAREM1 and SQUAREM2 showed poor performance, especially for the second set of starting values where SQUAREM1 failed with invalid solutions for 312 times; this is due to the reason explained in Section 3.5. Both DECME-2 and DECME-2s failed with invalid solutions occasionally, which should not have happened theoretically and is due to accumulated numerical errors along the boundary. In theory, an EM update of a valid estimate should always be valid. However, if the input value is close to boundary, numerical errors may cause the output of an EM iteration to fall outside of the constrained area.

In terms of number of EM iterations, DECME-2 is clearly the best and outperforms EM by factors of about 70 (116 versus $8,013$ on average) and 190 (299 versus $57,072$ on average) for the two sets of starting values. DECME-2 is also faster than the two SQUAREM methods by factors of more than 10 and more than 20 for the two sets of examples. This result is consistent with the theoretical result on the fast convergence of DECME-2. As expected, the number of iterations for DECME-2s (146 and 432 on average for the two sets of examples, respectively) did not increase much compared to DECME-2. Thus not surprisingly, DECME-2s is the best in terms of CPU time: it outperformed EM by a factor of about 16 and 40 on average under the two sets of starting values. DECME-2s is also about two and four times faster than SQUAREM for the two set of starting values.

A PX-EM implementation for a type of linear mixed-effects model has been proposed in Liu et al. (1998). However, it is not applicable to the current example, where the subjects are assumed to be from two different groups with different variances of the error terms.

In this case, each iteration of PX-EM needs to solve a weighted-least-square problem with unknown variances, which does not have a closed-from solution.

4.3. The Factor Analysis Example

Here we used the factor analysis example in Section 2.2, but with a minor modification. Note that in the original model, the first two rows of the loading matrix β are still subject to an orthogonal rotation. To make the comparison more reliable, we further enforced zero factor loading on factor 1 for variable 1. This modified model has 35 free parameters $\theta = (\beta_{1,2-9}, \beta_{2\cdot}, \beta_{3,1-4}, \beta_{4,5-9}, \sigma^2)'$.

The difficulty of using EM to fit a factor analysis model has been reported in a number of papers and at least three issues have been discussed. First, it is reported in Liu and Rubin (1998) that the extremely slow increase of loglikelihood far before convergence makes it difficult to assess the convergence of EM. Second, the possible presence of multiple local maximums have been discussed (Rubin and Thayer, 1982; Bentler and Tanaka, 1983; Duan and Simonato, 1993, *etc.*). Third, optimisation algorithms may converge to improper (Heywood) solutions, *i.e.,* certain components of the estimated uniqueness are close to zero (see Jöreskog, 1967; Duan and Simonato, 1993, and the references therein).

The PX-EM implementation for explanatory factor analysis proposed in Liu et al. (1998) cannot be directly applied for the current confirmatory factor analysis model because the parameter expanded model does not preserve the observed-data likelihood due to the constraints of zero factor loadings. To remedy this problem, we change the expansion parameter, the matrix α in Liu et al. (1998), to a diagonal matrix. The PX-EM implementation under this expanded model is obtained by replacing $\alpha^{(t+1)} = C_{zz}^{(t+1)}$ in Liu et al. (1998) with a diagonal matrix $\alpha^{(t+1)}$ where $\mathsf{diag}(\alpha^{(t+1)}) = \mathsf{diag}(C_{zz}^{(t+1)})$.

The initial value for each component of the unconstrained β was chosen from $U(-50,\ 50)$. Each component of the constrained parameter σ^2 was started from $U(0.1,\ 100)$. The results for the $5,000$ runs of this example are shown in Table 9. As shown by the number of "non-optimal" cases in Table 9, all algorithms stopped prematurely for a large proportion of runs. This happened most frequently for EM (1,486 out of 5,000 runs), whereas DECME-2s was the best (1,135 out of 5,000 runs). All the "successful" cases were terminated by criterion $C1$, i.e., the final loglikelihood estimations for all "successful" cases were essentially the same. Table 10 shows the total number of Heywood cases (#(Heywood)) and the number of Heywood cases among the "non-optimal" cases (#(Heywood | non)) for all seven algorithms. It is clear that the occurrence of "non-optimal" cases from DECME and SQUAREM were all due to the Heywood problem, *i.e.*, these algorithms stopped prematurely around the boundary. EM and PX-EM didn't show a similar pattern about their premature stops. One explanation is that both EM and PX-EM were so slow that they were terminated by criterion $C2$ or $C3$, as discussed in Liu and Rubin (1998). Furthermore, if we relax criterion for defining "successful" runs by increasing ϵ_g from 10^{-4} to 10^{-3}, a large number of "non-optimal" cases will enter the "successful" group (#(near optimal) in Table 10). For the four DECME and SQUAREM algorithms, all these transition cases belong to Heywood cases and their estimations are clearly not in the same neighborhood of the estimations from the "successful" cases.

In terms of efficiency, DECME-2 is the best in terms of convergence rate and it is about 30 times faster than EM on average. DECME-2s is the best and it outperforms EM by a factor of 10 on average in terms of CPU time. DECME-2s is also better than SQUAREM1 and SQUAREM2, in terms of both convergence rate (82 EM iterations versus 269 EM iterations) and CPU time (0.25 seconds per run versus about 0.3 seconds per run on average).

We note that PX-EM is only slightly more efficient than EM for this example as compared to its performance for the examples in Liu et al. (1998). As discussed in Liu et al. (1998), the fast convergence of PX-EM is due to the adjustment on the estimation for the deviations between $C_{zz}^{(t+1)}$ and its expectation under the original EM model. For the explanatory factor analysis model in Liu et al. (1998), the adjustment is performed with a matrix for which all elements are free parameters. In the current case, however, only a diagonal matrix can be used due to the constraints on the loading matrix. The use of the reduced parameter expansion space may well explain the slow convergence of PX-EM for the current example.

4.4. A Bivariate t Example with Unknown Degrees of Freedom

Consider the multivariate t distribution $t_p(\mu, \Psi, \nu)$ with mean μ, scale matrix Ψ, and degrees of freedom ν. Finding the MLE of the parameters (μ, Ψ, ν) is a well known interesting application of the EM-type algorithms. For the case of known degrees of freedom ν, a simple but effective strategy for accelerating EM has been developed from different perspectives in several papers (Kent et al., 1994; Meng and van Dyk, 1997; Liu et al., 1998). Extensions of EM for the case with unknown ν can be found in Liu and Rubin (1994), Liu and Rubin (1995), Liu (1997), and Meng and van Dyk (1997). One of AECMs of Meng and van Dyk (1997), called AECME 1 and recommended by its authors, was used here for comparison.

Here we used the bivariate t distribution example in Liu and Rubin (1994), where the data are adapted from Table 1 of Cohen et al. (1993). The initial value for each component of the unconstrained μ was chosen from $U(-50,\ 50)$. Each component of the constrained parameters $\Psi_{1,1}$, $\Psi_{2,2}$, and ν was started at a random draw from $U(0.1,\ 100)$, except for $\Psi_{1,2}$, which was started from 0. The results are shown in Table 11.

All algorithms showed nice stability for this example where all runs converged successfully. In terms of convergence rate, DECME-2 and DECME-2s again significantly outperformed the others, *e.g.,* DECME-2 was 10 times faster than EM even though EM has been relatively fast for this example (248 iterations on average). However, in terms of CPU time, SQUAREM1 and SQUAREM2 appeared to be slightly faster than DECME-2s since the latter performed more loglikelihood evaluations. From Table 11, AECM 1 only showed mild improvement over EM in terms of number of EM iterations (201 compared to 248 on average). In terms of CPU time, the two were almost the same because an iteration of AECM 1 is computationally more expensive than an EM iteration.

It is not surprising that SQUAREM may outperform DECME for problems where EM converges relatively fast as in the current example. The reason is that SQUAREM relies more heavily on the EM updates. Each iteration of SQUAREM runs three EM iterations, while DECME uses only one.

4.5. Gaussian Mixture Examples

The EM algorithm has been known as a powerful method for fitting mixture models, which are popular in many areas such as machine learning and pattern recognition (*e.g.,* Jordan and Jacobs, 1994; McLachlan and Krishnan, 1997; McLachlan and Peel, 2000; Bishop, 2006). While the slow convergence of EM has been frequently reported for fitting mixture models, a few acceleration methods have been proposed in the literature (Liu and Sun, 1997; Dasgupta and Schulman, 2000; Celeux et al., 2001; Pilla and Lindsay, 2001, among others). Here we show that DECME, as an off-the-shelf accelerator, can be easily applied to achieve dramatically faster convergence than EM.

Consider the class of mixtures of two univariate normal densities used by Redner and Walker (1984) to illustrate the relation between the efficiency of EM and the separation of the component populations in the mixture. The mixture has the form of

$$\begin{aligned} p(x|\pi_1,\pi_2,\mu_1,\mu_2,\sigma_1^2,\sigma_2^2) &= \pi_1 p_1(x|\mu_1,\sigma_1^2)+\pi_2 p_2(x|\mu_2,\sigma_2^2), \\ p_i(x|\mu_i,\sigma_i^2) &= \tfrac{1}{\sqrt{2\pi\sigma_i^2}} e^{-(x-\mu_i)^2/2\sigma_i^2},\ i=1,2. \end{aligned} \tag{9}$$

Let $\pi_1 = 0.3,\ \pi_2 = 0.7,\ \sigma_1^2 = \sigma_2^2 = 1$, and $\mu_1 = -\mu_2$. Random samples of $1,000$ observations were generated from each case of $\mu_1 - \mu_2 = 6,\ 4,\ 3,\ 2$, and 1.5. Since EM converges very fast for the first three cases, we used the last two cases for comparing different algorithms. For each of the two cases, $\mu_1 - \mu_2 = 2$ and 1.5, we simulated 5,000 data sets. For each simulated data set, all the algorithms were started from $\pi_1^{(0)} = \pi_2^{(0)} = 0.5,\ \sigma_i^{2^{(0)}} = 0.5$, and $\mu_i^{(0)} = 1.5\mu_i$ for $i = 1$ and 2. The algorithms were terminated when either $C2$ or $C3$ in (8) was satisfied. Criterion $C1$ was not applied for this example since it takes too long to obtain L_{max} for all data sets. The results are shown in Tables 12 and 13.

For the case of $\mu_1 - \mu_2 = 2$, all algorithms showed nice stability. However, for the $\mu_1 - \mu_2 = 1.5$ case where EM becomes very slow, the stability issue turned out to be a liability problem for EM and SQUAREM2. There were 261 EM runs stopped prematurely, and 57 SQUAREM2 runs stopped with invalid solutions. Although both situations also happened for DECME-2 and DECME-2s, they appeared to be less severe. It is noteworthy that 5 EM runs ended up with invalid solutions when $\mu_1 - \mu_2 = 1.5$, due to accumulated numerical errors when the estimation is close to the boundary of the feasible region; see, *e.g.,* Ueda et al. (2000) and Figueiredo and Jain (2002) for more discussion on this issue.

In terms of number of EM iterations, DECME-2 is still the best and outperforms EM by factors of about 80 and 115 on average for the two cases. In terms of CPU time, DECME-2s , SQUAREM1, and SQUAREM2 were very similar. They were about 10 and 20 times faster than EM for the two cases.

5. Conclusion

Dynamically constructing the subspaces for ML-steps of ECME is shown to be promising for accelerating the EM algorithm. We have formulated this idea into the generic DECME algorithm. An implementation of DECME, called DECME-2, is shown to be equivalent to a conjugate direction method, which provides theoretical justification for its fast convergence.

To improve efficiency in terms of CPU time, a variant of DECME-2, called DECME-2s, was proposed to remedy the problem of DECME-2that it uses too many loglikelihood evaluations in each iteration.

The numerical results show that DECME-2s typically outperforms EM by significant margins in terms of both convergence rate and CPU time. These examples also show that DECME-2s holds a clear edge over other state-of-art EM accelerating algorithms such as SQUAREM and PX-EM, especially when the model is complicated and EM is very slow. We note that although the statistical models used in the numerical studies has variuous structures, the number of unknown parameters are not very large. The computational gain of DEMCE over EM and other acceleration methods remains to be seen in the future. It should be also noted that, sharing with DECME the same fundamental idea of adaptively constructing acceleration subspaces, the SQUAREM algorithm shows impressive performance over EM.

Our main focus in the current paper has been on accelerating EM. However, it is noteworthy that the proof of Theorem 3.3 only depends on the linear convergence rate of the underlying algorithm being accelerated rather than its specific structure. Hence an immediate point to make is that the proposed methods should also work for other EM-type algorithms of linear convergence rate or more broadly for the MM algorithm (Hunter and Lange, 2004). We leave this problem open for future investigation.

Acknowledgement

The authors thank Mary Ellen Bock, Aiyou Chen, William Cleveland, Ahmed Sameh, David van Dyk, Hao Zhang, Heping Zhang, and Jian Zhang for helpful discussions and suggestions. They are especially grateful to the Editor, AE, and referees for their helpful and constructive comments and to Professor Ravi Varadhan for sending us the code of SQUAREM and the helpful discussions which led to a more meaningful comparison study. Yunxiao He's research was partially supported by NIH grant R01DA016750. Chuanhai Liu was partially supported by the U.S. National Science Foundation grant NSF-DMS-1007678.

Appendix

A. The Linear Convergence Rate of EM: a Quick Review

This section reviews some well known convergence properties of EM (see, *e.g.,* Dempster et al. (1977) and Meng and Rubin (1994)) to establish necessary notations. In a small neighbourhood of the MLE, the observed log-likelihood $L(\theta|Y_{obs})$ may be approximately by a quadratic function:

$$L(\theta|Y_{obs}) = -\frac{1}{2}(\theta - \hat{\theta})' I_{obs}(\theta - \hat{\theta}). \tag{10}$$

With this approximation, Dempster et al. (1977) proved that EM has a linear convergence rate determined by DM^{EM} in (1). The matrix DM^{EM} is called the missing information fraction. It is named after the following identity:

$$DM^{EM} = I_p - I_{com}^{-1} I_{obs} = I_{com}^{-1} I_{mis}, \tag{11}$$

where I_p represents the identity matrix of order p, I_{obs} and I_{com} are the negative Hessian matrices of $L(\theta|Y_{obs})$ and $Q(\theta|Y_{obs},\hat{\theta})$ at the MLE, and $I_{mis} = I_{com} - I_{obs}$. The matrices I_{obs}, I_{mis} and I_{com} are usually called observed-data, missing-data, and complete-data information matrices. We assume that these matrices are positive definite.

Since I_{com} is positive definite, there exists a positive definite matrix, denoted by $I_{com}^{1/2}$, such that $I_{com} = I_{com}^{1/2} I_{com}^{1/2}$. Denote by $I_{com}^{-1/2}$ the inverse of $I_{com}^{1/2}$. Then $I_{com}^{-1} I_{obs}$ is similar to $I_{com}^{1/2} \times I_{com}^{-1} I_{obs} \times I_{com}^{-1/2} = I_{com}^{-1/2} I_{obs} I_{com}^{-1/2}$ and, thereby, $I_{com}^{-1} I_{obs}$ and $I_{com}^{-1/2} I_{obs} I_{com}^{-1/2}$ have the same eigenvalues. Since $I_{com}^{-1/2} I_{obs} I_{com}^{-1/2}$ is symmetric, there exists an orthogonal matrix T such that

$$I_{com}^{-1/2} I_{obs} I_{com}^{-1/2} = T \Lambda T', \tag{12}$$

where $\Lambda = \mathsf{diag}(\lambda_1, \cdots, \lambda_p)$, and $\lambda_i,\ i = 1, \cdots, p$, are the eigenvalues of $I_{com}^{-1} I_{obs}$. Therefore,

$$I_{com}^{-1} I_{obs} = I_{com}^{-1/2} T \Lambda T' I_{com}^{1/2}. \tag{13}$$

Let $P = I_{com}^{-1/2} T$, then we have $I_{com}^{-1} I_{obs} = P \Lambda P^{-1}$. Furthermore, the columns of P and the rows of P^{-1} are eigenvectors of $I_{com}^{-1} I_{obs}$ and $I_{obs} I_{com}^{-1}$, respectively. Define $\eta = P^{-1}(\hat{\theta} - \theta)$, then from (1) we have $\eta_t = (I_p - \Lambda)\eta_{t-1}$, or equivalently

$$\eta_{t,i} = (1 - \lambda_i)\eta_{t-1,i},\ i = 1, \cdots, p. \tag{14}$$

Equation (14) implies that EM converges independently along the p eigenvector directions of $I_{com}^{-1} I_{obs}$ (or equivalently DM^{EM}) with the rates determined by the corresponding eigenvalues. For simplicity of the later discussion, we assume $1 > \lambda_1 > \lambda_2 > \cdots > \lambda_p > 0$ and $\eta_{0,i} \neq 0$ for $i = 1, \cdots, p$.

B. The Conservative Step Size of EM: Proof of Theorem 3.1

From (10) and the definition of DECME-1 in Section 3.2, it is easy to show that

$$\alpha_t = \frac{(\theta_t - \tilde{\theta}_{t-1})' I_{obs} (\hat{\theta} - \theta_t)}{(\theta_t - \tilde{\theta}_{t-1})' I_{obs} (\theta_t - \tilde{\theta}_{t-1})}. \tag{15}$$

Making use of the fact that $\theta_t = \theta_{t-1} + I_{com}^{-1} I_{obs} (\hat{\theta} - \theta_{t-1})$, followed from (1) and (11), leads to

$$\alpha_t = \frac{(\hat{\theta} - \tilde{\theta}_{t-1})' I_{obs} I_{com}^{-1} I_{obs} (\hat{\theta} - \tilde{\theta}_{t-1})}{(\hat{\theta} - \tilde{\theta}_{t-1})' I_{obs} I_{com}^{-1} I_{obs} I_{com}^{-1} I_{obs} (\hat{\theta} - \tilde{\theta}_{t-1})} - 1. \tag{16}$$

By definition of η, we have $\hat{\theta} - \tilde{\theta}_{t-1} = I_{com}^{-1/2} T \tilde{\eta}_{t-1}$. Making use of (12) and the fact that T is an orthogonal matrix yields

$$\alpha_t = \frac{\tilde{\eta}_{t-1}' \Lambda^2 \tilde{\eta}_{t-1}}{\tilde{\eta}_{t-1}' \Lambda^3 \tilde{\eta}_{t-1}} - 1. \tag{17}$$

Since Λ is diagonal and all its diagonal elements are between 0 and 1, it follows immediately that $\alpha_t > 0$. □

C. The Convergence of DECME-1: Proof of Theorem 3.2

Similar to (1) and (14) for EM, we have the following results for DECME-1:

$$\hat{\theta} - \tilde{\theta}_t = [I_p - (1+\alpha_t) I_{com}^{-1} I_{obs}](\hat{\theta} - \tilde{\theta}_{t-1}), \tag{18}$$

and

$$\tilde{\eta}_{t,i} = [1 - (1+\alpha_t)\lambda_i]\tilde{\eta}_{t-1,i} \ , \quad i = 1, \cdots, p. \tag{19}$$

For the $p = 2$ case, from (17) we have

$$\alpha_t = \frac{\lambda_1^2 \tilde{\eta}_{t-1,1}^2 + \lambda_2^2 \tilde{\eta}_{t-1,2}^2}{\lambda_1^3 \tilde{\eta}_{t-1,1}^2 + \lambda_2^3 \tilde{\eta}_{t-1,2}^2} - 1, \tag{20}$$

and then,

$$1 - (1+\alpha_t)\lambda_1 = \frac{\lambda_2^2(\lambda_2 - \lambda_1)\tilde{\eta}_{t-1,2}^2}{\lambda_1^3 \tilde{\eta}_{t-1,1}^2 + \lambda_2^3 \tilde{\eta}_{t-1,2}^2}, \ 1 - (1+\alpha_t)\lambda_2 = \frac{\lambda_1^2(\lambda_1 - \lambda_2)\tilde{\eta}_{t-1,1}^2}{\lambda_1^3 \tilde{\eta}_{t-1,1}^2 + \lambda_2^3 \tilde{\eta}_{t-1,2}^2}. \tag{21}$$

From (19) and (21), we get

$$\frac{\tilde{\eta}_{t,1}}{\tilde{\eta}_{t,2}} = -\frac{\lambda_2^2}{\lambda_1^2} \frac{\tilde{\eta}_{t-1,2}}{\tilde{\eta}_{t-1,1}}. \tag{22}$$

It follows that $\tilde{\eta}_{t,1}/\tilde{\eta}_{t,2} = \eta_{t-2,1}/\tilde{\eta}_{t-2,2}$. Furthermore, from (20) we have

$$\alpha_t = \frac{\lambda_1^2 (\tilde{\eta}_{t-1,1}/\tilde{\eta}_{t-1,2})^2 + \lambda_2^2}{\lambda_1^3 (\tilde{\eta}_{t-1,1}/\tilde{\eta}_{t-1,2})^2 + \lambda_2^3} - 1, \tag{23}$$

and immediately $\alpha_t = \alpha_{t-2}$ holds, which proves Conclusion 1.

Now consider a trivial algorithm, called SOR2, where each iteration of SOR2 includes two iterations of DECME-1 . From (18), we have

$$\hat{\theta} - \tilde{\theta}_{t+1} = [I_2 - (1+\alpha_t) I_{com}^{-1} I_{obs}][I_2 - (1+\alpha_{t-1}) I_{com}^{-1} I_{obs}](\hat{\theta} - \tilde{\theta}_{t-1}). \tag{24}$$

By conclusion 1, $[I_p - (1+\alpha_t) I_{com}^{-1} I_{obs}][I_p - (1+\alpha_{t-1}) I_{com}^{-1} I_{obs}]$ is a constant matrix, denoted by DM^{SOR2}, which obviously determines the convergence rate of SOR2. By using (13), we have $DM^{SOR2} = I_{com}^{-1/2} T [I_p - (1+\alpha_t)\Lambda][I_p - (1+\alpha_{t-1})\Lambda] T' I_{com}^{1/2}$. Moreover, with (22) and (23), it is easy to show that

$$[1 - (1+\alpha_t)\lambda_j][1 - (1+\alpha_{t-1})\lambda_j] = \frac{(\lambda_2 - \lambda_1)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2 \left(\frac{\lambda_1^2}{\lambda_2^2} \frac{\tilde{\eta}_{t-1,1}^2}{\tilde{\eta}_{t-1,2}^2} + \frac{\lambda_2^2}{\lambda_1^2} \frac{\tilde{\eta}_{t-1,2}^2}{\tilde{\eta}_{t-1,1}^2} \right)}, \ j = 1, 2. \tag{25}$$

It follows that $DM^{SOR2} = [1-(1+\alpha_t)\lambda_1][1-(1+\alpha_{t-1})\lambda_1] I_2$, which means SOR2 converges with the same rate $[1 - (1+\alpha_t)\lambda_1][1 - (1+\alpha_{t-1})\lambda_1]$ along any direction. From equation (25), it is easy to see that

$$[1 - (1+\alpha_t)\lambda_1][1 - (1+\alpha_{t-1})\lambda_1] \leq \frac{(\lambda_1 - \lambda_2)^2}{(\lambda_1 + \lambda_2)^2} = (1 - \frac{2\lambda_2}{\lambda_1 + \lambda_2})^2 < (1 - \lambda_2)^2.$$

Note that $(\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2)$ is the optimal convergence rate of SORF and that $1 - \lambda_2$ is the convergence rate of EM. Hence Conclusion 2 follows.

Since $\lambda_1 > \lambda_2$, (21) implies that $1-(1+\alpha_t)\lambda_1 < 0$ and $1-(1+\alpha_t)\lambda_2 > 0$. It follows from (19) that $\tilde{\eta}_{t,1}\tilde{\eta}_{t-1,1} < 0$ and $\tilde{\eta}_{t,2}\tilde{\eta}_{t-1,2} > 0$. This proves the first statement in conclusion 3. Note that $\tilde{\theta}_{t+1} - \tilde{\theta}_t = (I - DM^{SOR2})(\hat{\theta} - \tilde{\theta}_{t-1}) \propto \hat{\theta} - \tilde{\theta}_{t-1}$. Hence $\tilde{\theta}_{t+1} - \tilde{\theta}_{t-1}$ is parallel to $\hat{\theta} - \tilde{\theta}_{t-1}$, which concludes the second statement in Conclusion 3. □

D. The Convergence of DECME-2: Proof of Theorem 3.3

We prove Theorem 3.3 by induction. The proof is similar to that of the PARTAN theorem in Luenberger (2003, pp. 255-256). However, the difference between a generalised gradient direction and the gradient direction needs to be taken into account.

It is certainly true for $t = 1$ since the first iteration is a line search along the EM direction for both DECME-2 and AEM.

Now suppose that $\tilde{\theta}_0, \tilde{\theta}_1, \cdots, \tilde{\theta}_{t-1}$ have been generated by AEM and $\tilde{\theta}_t$ is determined by DECME-2. We want to show that $\tilde{\theta}_t$ is the same point as that generated by another iteration of AEM. For this to be true $\tilde{\theta}_t$ must be the point that maximises $L(\theta|Y_{obs})$ over the two-dimensional plane $\tilde{\theta}_{t-1} + \{\tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}, \theta_t - \tilde{\theta}_{t-1}\}$. Since $L(\theta|Y_{obs})$ is assumed to be quadratic with a positive definite Hessian matrix, $L(\theta|Y_{obs})$ is strictly convex and we only need to prove $\tilde{g}_t$ (gradient of $L(\theta|Y_{obs})$ at $\tilde{\theta}_t$) is orthogonal to $\tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}$ and $\theta_t - \tilde{\theta}_{t-1}$, or equivalently $\tilde{\theta}^\star_t - \tilde{\theta}_{t-2}$ and $\theta_t - \tilde{\theta}_{t-1}$. Since $\tilde{\theta}_t$ maximises $L(\theta|Y_{obs})$ along $\tilde{\theta}^\star_t - \tilde{\theta}_{t-2}$, $\tilde{g}_t$ is orthogonal to $\tilde{\theta}^\star_t - \tilde{\theta}_{t-2}$. Similarly, $\tilde{g}^\star_t$ is orthogonal to $\theta_t - \tilde{\theta}_{t-1}$. Furthermore, we have $\tilde{g}'_{t-2}(\theta_t - \tilde{\theta}_{t-1}) = (\hat{\theta} - \tilde{\theta}_{t-2})' I_{obs} I^{-1}_{com} I_{obs}(\hat{\theta} - \tilde{\theta}_{t-1}) = (\theta_{t-1} - \tilde{\theta}_{t-2})'\tilde{g}_{t-1} = 0$, where the last identity is true due to the Expanding Subspace Theorem (Luenberger, 2003, p. 241) for the conjugate direction methods. Then $\tilde{g}'_t(\theta^\star_t - \tilde{\theta}_{t-1}) = (\hat{\theta} - \tilde{\theta}_t)' I_{obs}(\theta^\star_t - \tilde{\theta}_{t-1}) = [\hat{\theta} - \tilde{\theta}_{t-2} - (1 + \alpha^{(2)}_t)(\tilde{\theta}^\star_t - \tilde{\theta}_{t-2})]' I_{obs}(\theta^\star_t - \tilde{\theta}_{t-1}) = [-\alpha^{(2)}_t(\hat{\theta} - \tilde{\theta}_{t-2}) I_{obs} + (1 + \alpha^{(2)}_t)(\hat{\theta} - \tilde{\theta}^\star_t)' I_{obs}]'(\theta^\star_t - \tilde{\theta}_{t-1}) = [-\alpha^{(2)}_t \tilde{g}_{t-2} + (1 + \alpha^{(2)}_t)\tilde{g}^\star_t]'(\theta^\star_t - \tilde{\theta}_{t-1}) = 0$. It follows that $\tilde{g}_t$ is orthogonal to $\theta^\star_t - \tilde{\theta}_{t-1}$. □

E. Computation of the Feasible Region

As discussed in Sections 3.3 and 3.4, both DECME-2 and DECME-2s rely on the computation of the feasible region given the current estimate θ and a searching direction d. For the constraints involved in our numerical examples, we summarise the methods to obtain the feasible region as follows. If a model has several sets of constraints, we can determine the feasible region induced by each of them and then take their intersection. Without loss of generality, we assume in the following that d is the counterpart of the discussed parameters in the vector representing the search direction.

1.) The degrees of freedom ν in the t distribution. It is easy to compute the feasible region by solving $\nu + \alpha\mathsf{d} > 0$.

2.) The mixing coefficients, $\pi_i,\ i = 1, \cdots, K$, in the mixture model. There are two types of constraints here, *i.e.,* $\sum_{i=1}^K \pi_i = 1$ and $\pi_i \geq 0$. The first constraint is equivalent to that the first $K - 1$ coefficients with constraints $\sum_{i=1}^{K-1} \pi_i \leq 1$ and $\pi_i \geq 0$ for

$i = 1, \cdots, K-1$. Then the feasible region is given by the intersection of the solutions for the inequalities $\sum_{i=1}^{K-1} \pi_i + \alpha \sum_{i=1}^{K-1} d_i \leq 1$ and $\pi_i + \alpha d_i \geq 0,\ i = 1, \cdots, K-1$.

3.) The variance components in the linear mixed-effects model and the mixture model, and the uniquenesses in the factor analysis model. This can be handled in the same way as that for the degrees of freedom in the t distribution.

4.) The covariance matrices in the linear mixed-effects model and the t distribution. For the current paper, only two-dimensional covariance matrices are involved. A two-dimensional matrix Ψ is positive definite if and only if $\Psi_{1,1} > 0$ and $\det(\Psi) > 0$. Hence we only need to guarantee $\Psi_{1,1} + \alpha d_{1,1} > 0$ and $\det(\Psi + \alpha D) > 0$ (assume D is the matrix generated from the vector d in the same way as Ψ is generated from θ). For other covariance matrices of fairly small size, a similar method could be used. When the dimension of the covariance matrix is high, it is a common practise to enforce certain structure on the matrix. For example, in spatial statistics, the covariance matrices are usually assumed to be generated from various covariance functions with very few parameters (Zhang, 2002; Zhu et al., 2005; Zhang, 2007). The feasible region of α can be easily obtained.

Note that the intervals computed above are usually very wide and some other information may be used to narrow them down. For example, we always started the line search by forcing $\alpha > -1$ in our implementation. In addition, to prevent the estimation from getting too close to the boundary, some simple tricks can be used. For example, in t distribution, instead of solving $\nu + \alpha \mathsf{d} > 0$, we may solve $\nu + \alpha \mathsf{d} > \epsilon$ for a very small positive number ϵ.

References

Bentler, P. and J. Tanaka (1983). Problems with EM algorithms for ML factor analysis. *P*sychometrika *4*8(2), 247–251.

Bishop, C. M. (2006). *P*attern recognition and machine learning. Information Science and Statistics. New York: Springer.

Celeux, G., S. Chrétien, F. Forbes, and A. Mkhadri (2001). A component-wise EM algorithm for mixtures. *J.* Comput. Graph. Statist. *1*0(4), 697–712.

Cohen, M., S. R. Dalal, and J. W. Tukey (1993). Robust, smoothly heterogeneous variance regression. *J*ournal of the Royal Statistical Society, Series C: Applied Statistics *4*2, 339–353.

Concus, P., G. H. Golub, and D. P. O'Leary (1976). A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In *S*parse matrix computations (Proc. Sympos., Argonne Nat. Lab., Lemont, Ill., 1975), pp. 309–332. New York: Academic Press.

Dasgupta, S. and L. J. Schulman (2000). A two-round variant of em for gaussian mixtures. In *U*AI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, USA, pp. 152–159. Morgan Kaufmann Publishers Inc.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *J*ournal of the Royal Statistical Society, Series B: Methodological *3*9, 1–22.

Duan, J. and J. Simonato (1993). Multiplicity of solutions in maximum likelihood factor analysis. *J*ournal of Statistical Computation and Simulation *4*7(1), 37–47.

Figueiredo, M. and A. Jain (2002). Unsupervised learning of finite mixture models. *I*EEE Transactions on pattern analysis and machine intelligence *2*4(3), 381–396.

Fletcher, R. and C. M. Reeves (1964). Function minimization by conjugate gradients. *C*omput. J. *7*, 149–154.

Frankel, S. (1950). Convergence rates of iterative treatments of partial differential equations. *M*athematical Tables and Other Aids to Computation *4* (30), 65–75.

Gelfand, A. E., S. E. Hills, A. Racine-Poon, and A. F. M. Smith (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *J*ournal of the American Statistical Association *8*5, 972–985.

Golub, G. H. and S. G. Nash (1982). Nonorthogonal analysis of variance using a generalized conjugate-gradient algorithm. *J*. Amer. Statist. Assoc. *7*7(377), 109–116.

He, Y. (2009). Improving the EM algorithm for maximum likelihood inference. *E*TD Collection for Purdue University.

Hestenes, M. R. and E. Stiefel (1952). Methods of conjugate gradients for solving linear systems. *J*. Research Nat. Bur. Standards *4*9, 409–436 (1953).

Hesterberg, T. (2005). Staggered Aitken acceleration for EM. In *A*SA Proceedings of the Joint Statistical Meetings, pp. 2101–2110. American Statistical Association.

Hunter, D. and K. Lange (2004). A Tutorial on MM Algorithms. *T*he American Statistician *5*8(1), 30–38.

Jamshidian, M. and R. I. Jennrich (1993). Conjugate gradient acceleration of the em algorithm. *J*ournal of the American Statistical Association *8*8(421), 221–228.

Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the em algorithm. *N*eural Computation *6*, 181–214.

Jöreskog, K. (1967). Some contributions to maximum likelihood factor analysis. *P*sychometrika *3*2(4), 443–482.

Jöreskog, K. (1969, June). A general approach to confirmatory maximum likelihood factor analysis. *P*sychometrika *3*4(2), 183–202.

Kent, J., D. Tyler, and Y. Vard (1994). A curious likelihood identity for the multivariate t-distribution. *C*ommunications in Statistics-Simulation and Computation *2*3(2), 441–453.

Kowalski, J., X. Tu, R. Day, and J. Mendoza-Blanco (1997). On the rate of convergence of the ECME algorithm for multiple regression models with t-distributed errors. *B*iometrika *8*4(2), 269.

Laird, N. M. and J. H. Ware (1982). Random-effects models for longitudinal data. *B*iometrics *3*8, 963–974.

Lange, K. (1995). A gradient algorithm locally equivalent to the EM algorithm. *J*. Roy. Statist. Soc. Ser. B *5*7(2), 425–437.

Liu, C. (1997). ML estimation of the multivariate t distribution and the EM algorithm. *J*ournal of Multivariate Analysis *6*3, 296–312.

Liu, C. (1998). Information matrix computation from conditional information via normal approximation. *B*iometrika *8*5, 973–979.

Liu, C. and D. B. Rubin (1994). The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *B*iometrika *8*1, 633–648.

Liu, C. and D. B. Rubin (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. *S*tatistica Sinica *5*, 19–39.

Liu, C. and D. B. Rubin (1998). Maximum likelihood estimation of factor analysis using the ECME algorithm with complete and incomplete data. *S*tatist. Sinica *8*(3), 729–747.

Liu, C., D. B. Rubin, and Y. N. Wu (1998). Parameter expansion to accelerate EM: The PX-EM algorithm. *B*iometrika *8*5, 755–770.

Liu, C. and D. X. Sun (1997). Acceleration of EM algorithm for mixture models using ECME. In *A*SA Proceedings of the Statistical Computing Section, pp. 109–114. American Statistical Association.

Luenberger, D. (2003). *L*inear and Nonlinear Programming (2nd ed.). Springer.

McLachlan, G. and D. Peel (2000). *F*inite mixture models. Wiley Series in Probability and Statistics: Applied Probability and Statistics. Wiley-Interscience, New York.

McLachlan, G. J. and T. Krishnan (1997). *T*he EM algorithm and extensions. Wiley Series in Probability and Statistics: Applied Probability and Statistics. New York: John Wiley & Sons Inc. A Wiley-Interscience Publication.

Meng, X. and D. B. Rubin (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *B*iometrika *8*0, 267–278.

Meng, X. and D. B. Rubin (1994). On the global and componentwise rates of convergence of the EM algorithm (STMA V36 1300). *L*inear Algebra and its Applications *1*99, 413–425.

Meng, X. and D. van Dyk (1997). The EM algorithm – An old folk-song sung to a fast new tune (Disc: P541-567). *J*ournal of the Royal Statistical Society, Series B: Methodological *5*9, 511–540.

Pilla, R. S. and B. G. Lindsay (2001). Alternative EM methods for nonparametric finite mixture models. *B*iometrika *8*8(2), 535–550.

Pinheiro, J. C., C. Liu, and Y. Wu (2001). Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *J*ournal of Computational and Graphical Statistics *1*0(2), 249–276.

R Development Core Team (2010). *R*: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Redner, R. A. and H. F. Walker (1984). Mixture densities, maximum likelihood and the EM algorithm. *S*IAM Review *2*6, 195–202.

Roland, C. (2010). A note on the parameterized em method. *S*tatistics & Probability Letters *8*0.

Rubin, D. B. and D. T. Thayer (1982). EM algorithms for ML factor analysis. *P*sychometrika *4*7, 69–76.

Salakhutdinov, R. and S. Roweis (2003). Adaptive overrelaxed bound optimization methods. In *I*n Proceedings of International Conference on Machine Learning, ICML. International Conference on Machine Learning, ICML, pp. 664–671.

Sammel, M. and L. Ryan (1996). Latent variable models with fixed effects. *B*iometrics *5*2(2), 650–663.

Shah, B. V., R. J. Buehler, and O. Kempthorne (1964). Some algorithms for minimizing a function of several variables. *J*. Soc. Indust. Appl. Math. *1*2, 74–92.

Ueda, N., R. Nakano, Z. Ghahramani, and G. Hinton (2000). SMEM algorithm for mixture models. *N*eural computation *1*2(9), 2109–2128.

Varadhan, R. and C. Roland (2008). Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *S*cand. J. Statist. *3*5(2), 335–353.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *T*he Annals of Statistics *1*1, 95–103.

Young, D. (1954). Iterative methods for solving partial difference equations of elliptic type. *T*ransactions of the American Mathematical Society *7*6(1), 92–111.

Zhang, H. (2002). On estimation and prediction for spatial generalized linear mixed models. *B*iometrics *5*8(1), 129–136.

Zhang, H. (2007). Maximum-likelihood estimation for multivariate spatial linear coregionalization models. *E*nvironMetrics *1*8(2), 125–139.

Zhu, J., J. C. Eickhoff, and P. Yan (2005). Generalized linear latent variable models for repeated measures of spatially correlated multivariate data. *B*iometrics *6*1(3), 674–683.

Table 1. Eigenvalues of DM^{EM} and DM^{ECME} for the Linear Mixed-effects Example in Section 2.1

Algorithm	Eigenvalues of the missing information fraction
EM	0.9860 0.9746 0.7888 0.6706 0.5176 0.3874 0.3260 0.2710 0.0364
ECME	0.5176 0.3874 0.3260 0.2710 0.0364 0.0000 0.0000 0.0000 0.0000

Table 2. The Four Largest Eigenvalues and the Corresponding Eigenvectors of DM^{EM} for the Linear Mixed-effects Example in Section 2.1

Eigenvalue	Corresponding eigenvector
0.9860	(0.0000 0.0000 -0.0413 -0.9991 0.0000 0.0000 0.0000 0.0000 0.0000)
0.9746	(0.0413 0.9991 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000)
0.7888	(0.0000 0.0000 -0.0433 0.9991 0.0000 0.0000 0.0000 0.0000 0.0000)
0.6706	(-0.0433 0.9991 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000)

Table 3. The Ten Leading Eigenvalues of DM^{EM} and DM^{ECME} for the Factor Analysis Example in Section 2.2

Algorithm	Ten leading eigenvalues of the missing information fraction
EM	1-2E-12 0.9992 0.9651 0.9492 0.9318 0.8972 0.8699 0.8232 0.8197 0.7876
ECME-1	1-2E-12 0.9979 0.9509 0.9292 0.9124 0.8725 0.8480 0.8031 0.7877 0.7539
ECME-2	0.9987 0.8715 0.7321 0.6673 0.5184 0.4770 0.4496 0.3727 0.3369 0.0000

Table 4. The Two Largest Eigenvalues and the Corresponding Eigenvectors of DM^{EM} for the Factor Analysis Example in Section 2.2

Eigenvalue	Corresponding eigenvector
1-2E-12	0.0812 0.0934 -0.4897 -0.1335 0.0684 0.0748 0.0363 -0.0864 -0.0949
	0.0996 0.1288 0.7171 0.1962 0.1085 0.1138 0.0954 0.2099 0.2047
	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.9992	0.0046 0.0057 -0.0047 -0.0005 -0.0047 -0.0034 -0.0038 -0.0013 0.0005
	-0.0062 -0.0071 -0.0092 -0.0064 0.0053 0.0060 0.0045 0.0049 0.0034
	0.0267 0.0441 -0.2871 0.0013 -0.0103 -0.0177 -0.0106 -0.0139 -0.0144
	-0.0028 0.0079 -0.9557 0.0111 0.0013 0.0040 -0.0011 -0.0028 0.0006

Table 5. Psudocode for DECME-2

Iterate until convergence: t^{th} iteration of DECME
For $\mathsf{mod}(t-1,p) \neq 0$, given $\tilde{\theta}_{t-1}$ and $\tilde{\theta}_{t-2}$
1. $\theta_t = \mathsf{EM}(\tilde{\theta}_{t-1})$
2. Compute $\tilde{\theta}_t^\star$ by a line search over the subspace $\theta_t + \{\theta_t - \tilde{\theta}_{t-1}\}$
3. Compute $\tilde{\theta}_t$ by a line search over the subspace $\tilde{\theta}_t^\star + \{\tilde{\theta}_t^\star - \tilde{\theta}_{t-2}\}$
4. Check for convergence: $C2$ and $C3$ in equation (8)
For $\mathsf{mod}(t-1,p) = 0$ and given $\tilde{\theta}_{t-1}$: restart iteration
Only Steps 1, 2, 4 are run and set $\tilde{\theta}_t = \tilde{\theta}_t^\star$

Table 6. Psudocode for DECME-2s

Iterate until convergence: t^{th} iteration of DECME-2s
For $t = 1$ and given the starting value $\tilde{\theta}_0$
Compute $\tilde{\theta}_1 = \mathsf{EM}(\tilde{\theta}_0)$, $L(\tilde{\theta}_0 \vert Y_{obs})$, and $L(\tilde{\theta}_1 \vert Y_{obs})$
For $t > 1$, given $\tilde{\theta}_{t-1}$, $\tilde{\theta}_{t-2}$, $L(\tilde{\theta}_{t-1} \vert Y_{obs})$, and $L(\tilde{\theta}_{t-2} \vert Y_{obs})$
1. Compute $\theta_t = \mathsf{EM}(\tilde{\theta}_{t-1})$ and $L(\theta_t \vert Y_{obs})$
2. Compute the three directions $\mathsf{d}_t^{(1)} = \theta_t - \tilde{\theta}_{t-1}$, $\mathsf{d}_t^{(2)} = \theta_t - \tilde{\theta}_{t-2}$, and $\mathsf{d}_t^{(3)} = \tilde{\theta}_{t-1} - \tilde{\theta}_{t-2}$
3. Compute the maximum feasible step sizes $\bar{\alpha}_t^{(i)},\ i = 1,2,3$ (see appendix E)
4. Compute $\xi_t^{(i)} = \theta_t + \alpha_t^{(i)} \mathsf{d}_t^{(i)}$ and $L(\xi_t^{(i)} \vert Y_{obs})$, where $\alpha_t^{(i)} = \mathsf{min}(1,\ 0.9\bar{\alpha}_t^{(i)}),\ i = 1,2,3$
5. Compute $(\hat{x}_t, \hat{y}_t)$ (see equation 5)
6. Compute $\mathsf{d}_t^{(4)} = \hat{x}_t(\theta_t - \tilde{\theta}_{t-1}) + \hat{y}_t(\theta_t - \tilde{\theta}_{t-2})$ and the maximum feasible step size $\bar{\alpha}_t^{(4)}$.
7. Compute $\xi_t^{(4)} = \theta_t + \alpha_t^{(4)} \mathsf{d}_t^{(4)}$, where $\alpha_t^{(4)} = \mathsf{min}(1,\ 0.9\bar{\alpha}_t^{(4)})$.
8. Let $\tilde{\theta}_t$ be the point with maximum loglikelihood among $\theta_t,\ \xi_t^{(1)},\ \xi_t^{(2)},\ \xi_t^{(3)}$, and $\xi_t^{(4)}$
9. Check for convergence: $C2$ and $C3$ in equation (8)

Table 7. Comparison for the Linear Mixed-effects Example in Section 4.2 : First Set of Starting Values

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	**5000/0/0**	8013 (4844)/5000	8013 (4844)/5000	7.460 (4.600)/5000
DECME-2	4967/0/33	**116 (38)/4967**	2600 (877)/4967	0.999 (0.371)/4967
DECME-2s	4986/0/14	146 (51)/4977	730 (256)/4986	**0.453 (0.178)/4981**
SQUAREM1	4941/2/57	1236 (767)/4939	451 (274)/4939	0.968 (0.613)/4939
SQUAREM2	4901/3/96	1191 (718)/4899	401 (241)/4899	0.925 (0.573)/4899

Table 8. Comparison for the Linear Mixed-effects Example in Section 4.2 : Second Set of Starting Values

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	**4996/4/0**	57072 (40204)/4996	57072 (40204)/4996	54.116 (39.615)/4992
DECME-2	4991/0/9	**299 (112)/4991**	6871 (2624)/4991	2.720 (1.186)/4991
DECME-2s	4994/0/6	431 (173)/4992	2154 (868)/4994	**1.372 (0.626)/4993**
SQUAREM1	4679/9/312	7501 (5497)/4670	2847 (2153)/4659	5.995 (4.502)/4663
SQUAREM2	4898/6/96	6936 (5100)/4892	2334 (1714)/4892	5.472 (4.137)/4887

Table 9. Comparison for the Factor Analysis Example in Section 4.3

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	3514/1486/0	2498 (874)/3284	2498 (874)/3284	2.908 (1.097)/3287
PXEM	3808/1192/0	2159 (463)/3793	2159 (463)/3793	2.705 (0.722)/3798
DECME-2	3839/1161/0	**70 (26)/3464**	1563 (631)/3495	0.465 (0.222)/3491
DECME-2s	**3865/1135/0**	82 (40)/3606	411 (203)/3621	**0.246 (0.188)/3671**
SQUAREM1	3692/1308/0	269 (101)/3399	91 (34)/3399	0.303 (0.175)/3447
SQUAREM2	3694/1305/1	269 (101)/3399	91 (34)/3399	0.310 (0.184)/3461

Table 10. Diagnosis for the Factor Analysis Example in Section 4.3

Algorithm	#(Heywood)	#(Heywood | non)	#(near optimal)
EM	24	24	475
PXEM	0	0	80
DECME-2	1161	1161	332
DECME-2s	1135	1135	349
SQUAREM1	1308	1308	404
SQUAREM2	1305	1305	403

Table 11. Comparison for the Bivariate Student-t Example in Section 4.4

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	5000/0/0	248 (6)/4984	248 (6)/4984	0.211 (0.041)/4982
AECM 1	5000/0/0	201 (2)/4932	201 (2)/4932	0.200 (0.042)/4977
DECME-2	5000/0/0	**24 (2)/4981**	531 (50)/4959	0.123 (0.027)/4919
DECME-2s	5000/0/0	30 (3)/4990	149 (14)/4990	0.065 (0.019)/4923
SQUAREM1	5000/0/0	67 (19)/4972	25 (6)/4959	**0.043 (0.018)/4837**
SQUAREM2	5000/0/0	78 (23)/4900	27 (8)/4880	0.055 (0.027)/4857

Table 12. Comparison for the Gaussian Mixture Example in Section 4.5 : $\mu_1 - \mu_2 = 2$

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	4997/3/0	2449 (1704)/4891	\	1.759 (1.179)/4910
DECME-2	5000/0/0	**31 (9)/4918**	786 (278)/4940	0.367 (0.131)/4951
DECME-2s	5000/0/0	40 (15)/4939	194 (75)/4939	0.143 (0.061)/4971
SQUAREM1	5000/0/0	148 (73)/4928	50 (24)/4928	0.155 (0.079)/4930
SQUAREM2	4999/0/1	148 (72)/4924	\	**0.130(0.066)/4928**

Table 13. Comparison for the Gaussian Mixture Example in Section 4.5 : $\mu_1 - \mu_2 = 1.5$

Algorithm	#(suc/non/inv)	#(EM)/#(obs)	#(llk)/#(obs)	CPU(s)/#(obs)
EM	4734/261/5	7351 (6636)/4734	\	4.664 (3.889)/4690
DECME-2	**4988/6/6**	**56 (26)/4848**	1388 (662)/4842	0.633 (0.300)/4859
DECME-2s	4983/10/7	64 (30)/4916	317 (149)/4916	0.225 (0.106)/4924
SQUAREM1	4960/33/7	263 (158)/4822	89 (53)/4821	0.263 (0.157)/4834
SQUAREM2	4908/35/57	263 (157)/4770	\	**0.222 (0.135)/4791**

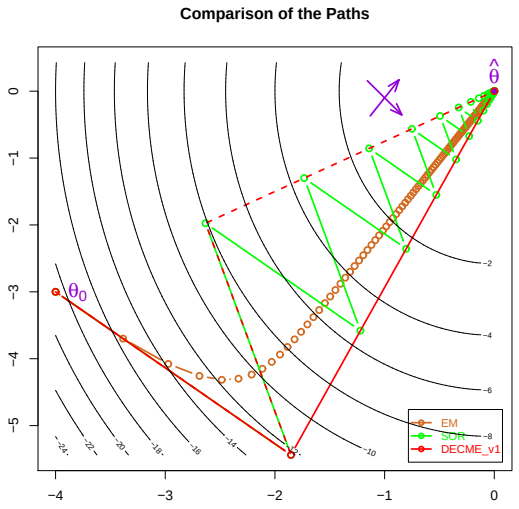

Fig. 1. Comparison of the Paths of EM, DECME-1, and DECME-2 for a Two-dimensional Example. The eigenvalues of DM^{EM} are 0.9684 and 0.6232; the darkviolet cross on the upright corner shows the directions of the two eigenvectors of DM^{EM}; The red dashed lines represent the true path of DECME-2 in its second iteration.

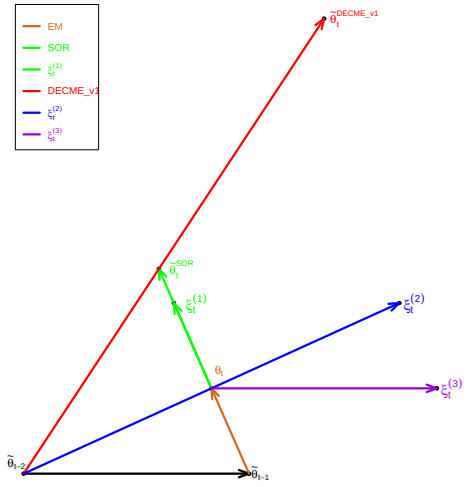

Fig. 2. Illustration for One Iteration of the DECME Implementations.

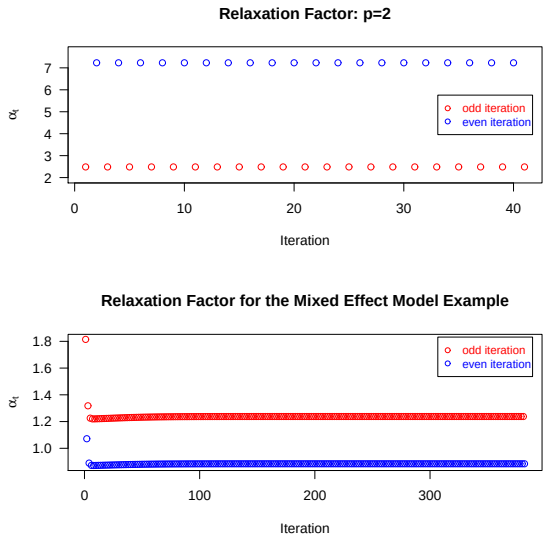

Fig. 3. Relaxation Factor α_t Generated from DECME-1. The top panel shows the sequence of α_t for the two-dimensional example used to generate Figure 1, and the bottom panel shows the sequence of α_t from the simulated nine-dimensional example in Section 3.2 by using information from the linear mixed-effects model example in Section 2.1 and Section 4.2 .