Stat Comput
DOI 10.1007/s11222-009-9166-3

A quasi-Newton acceleration for high-dimensional optimization algorithms

Hua Zhou · David Alexander · Kenneth Lange

Received: 2 June 2009 / Accepted: 30 November 2009

Abstract In many statistical problems, maximum likelihood estimation by an EM or MM algorithm suffers from excruciatingly slow convergence. This tendency limits the application of these algorithms to modern high-dimensional problems in data mining, genomics, and imaging. Unfortunately, most existing acceleration techniques are ill-suited to complicated models involving large numbers of parameters. The squared iterative methods (SQUAREM) recently proposed by Varadhan and Roland constitute one notable exception. This paper presents a new quasi-Newton acceleration scheme that requires only modest increments in computation per iteration and overall storage and rivals or surpasses the performance of SQUAREM on several representative test problems.

Keywords Maximum likelihood · Multivariate *t* · Admixture models · Imaging · Generalized eigenvalues

1 Introduction

Maximum likelihood and least squares are the dominant estimation methods in applied statistics. Because closed-form solutions to the score equations of maximum likelihood are the exception rather than the rule, numerical methods such as the EM algorithm (Dempster et al. 1977; Little and Rubin 2002; McLachlan and Krishnan 2008) enjoy wide usage. In the past decade, statisticians have come to realize that the EM algorithm is a special case of a broader MM (minorization-maximization or majorization-minimization) algorithm (de Leeuw 1994; Heiser 1995; Becker et al. 1997; Lange 2000; Hunter and Lange 2004; Wu and Lange 2008). This has opened new avenues to algorithm design. One advantage of MM algorithms is their numerical stability. Every MM algorithm heads uphill in maximization. In addition to this desirable ascent property, the MM algorithm handles parameter constraints gracefully. Constraint satisfaction is by definition built into the solution of the maximization step. However, MM algorithms suffer from two drawbacks. One is their often slow rate of convergence in a neighborhood of the maximum point. Slow convergence is an overriding concern in high-dimensional applications. A second criticism, which applies to scoring and Newton's method as well, is their inability to distinguish local from global maxima.

Most of the existing literature on accelerating EM algorithms is summarized in Chap. 4 of McLachlan and Krishnan (2008). As noted by Varadhan and Roland (2008), the existing methods can be broadly grouped into two categories. Members of the first category use the EM iterates to better approximate the observed information matrix of the log-likelihood. Examples include quasi-Newton approximation (Lange 95; Jamshidian and Jennrich 1997) and conjugate gradient methods (Jamshidian and Jennrich 1993). In exchange for speed, these methods sacrifice the stability and simplicity of the unadorned EM algorithm. The second category focuses on directly modifying a particular EM algorithm. These methods include data augmenta-

H. Zhou (✉)
Department of Human Genetics, University of California,
Los Angeles, CA, USA 90095
e-mail: huazhou@ucla.edu

D. Alexander
Department of Biomathematics, University of California,
Los Angeles, CA, USA

K. Lange
Departments of Biomathematics, Human Genetics, and Statistics,
University of California, Los Angeles, CA, USA

Published online: 12 December 2009

tion (Meng and van Dyk 1997), parameter expansion EM (PX-EM) (Liu et al. 1998), ECM (Meng and Rubin 1993), and ECME (Liu and Rubin 1994). Methods in the second category retain the ascent property of the EM algorithm while boosting its rate of convergence. However, they are ad hoc and subtle to concoct. Recently Varadhan and Roland (2008) have added a third class of squared iterative methods (SQUAREM) that seek to approximate Newton's method for finding a fixed point of the EM algorithm map. They resemble the multivariate version of Aitken's acceleration method (Louis 1982) in largely ignoring the observed log-likelihood. SQUAREM algorithms maintain the simplicity of the original EM algorithms, have minimal storage requirement, and are suited to high-dimensional problems.

In this article, we develop a new quasi-Newton acceleration that resembles SQUAREM in many respects. First, it is off-the-shelf and broadly applies to any search algorithm defined by a smooth algorithm map. It requires nothing more than the map updates and a little extra computer code. Second, in contrast to the previous quasi-Newton acceleration methods (Lange 1995; Jamshidian and Jennrich 1997), it neither stores nor manipulates the observed information matrix or the Hessian of the algorithm map. This makes it particularly appealing in high-dimensional problems. Third, although it does not guarantee the ascent property when applied to an MM algorithm, one can revert to ordinary MM whenever necessary. This fallback position is the major reason we focus on MM and EM algorithms. Algorithms such as block relaxation (de Leeuw 1994) and steepest ascent with a line search share the ascent property; these algorithms also adapt well to acceleration.

In Sect. 2, we describe the new quasi-Newton acceleration. Section 3 illustrates the basic theory by a variety of numerical examples. Our examples include the truncated beta-binomial model, a Poisson admixture model, the multivariate t distribution, an admixture model in genetics, PET imaging, a movie rating model, and an iterative algorithm for finding the largest or smallest generalized eigenvalue of a pair of symmetric matrices. The number of parameters ranges from two to tens of thousands in these examples. Our discussion summarizes our findings.

2 A quasi-Newton acceleration method

In this section we derive a new quasi-Newton method of acceleration for smooth optimization algorithms. Previous work (Lange 1995; Jamshidian and Jennrich 1997) takes up the current theme from the perspective of optimizing the objective function by Newton's method. This requires storing and handling the full approximate Hessian matrix, a demanding task in high-dimensional problems. It is also possible to apply Newton's method for finding a root of the equation $\mathbf{0} = x - F(x)$, where F is the algorithm map. This alternative perspective has the advantage of dealing directly with the iterates of the algorithm. Let $G(x)$ now denote the difference $G(x) = x - F(x)$. Because $G(x)$ has the differential $dG(x) = I - dF(x)$, Newton's method iterates according to

$$\begin{aligned} x^{n+1} &= x^n - dG(x^n)^{-1}G(x^n) \\ &= x^n - [I - dF(x^n)]^{-1}G(x^n). \end{aligned} \tag{1}$$

If we can approximate $dF(x^n)$ by a low-rank matrix M, then we can replace $I - dF(x^n)$ by $I - M$ and explicitly form the inverse $(I - M)^{-1}$.

Quasi-Newton methods operate by secant approximations. We can generate one of these by taking two iterates of the algorithm starting from the current point x^n. If we are close to the optimal point x^∞, then we have the linear approximation

$$F \circ F(x^n) - F(x^n) \approx M[F(x^n) - x^n],$$

where $M = dF(x^\infty)$. If v is the vector $F \circ F(x^n) - F(x^n)$ and u is the vector $F(x^n) - x^n$, then the secant requirement is $Mu = v$. In fact, for the best results we require several secant approximations $Mu_i = v_i$ for $i = 1, \ldots, q$. These can be generated at the current iterate x^n and the previous $q - 1$ iterates. One answer to the question of how to approximate M is given by the following proposition.

Proposition 1 *Let $M = (m_{ij})$ be a $p \times p$ matrix, and denote its squared Frobenius norm by $\|M\|_F^2 = \sum_i \sum_j m_{ij}^2$. Write the secant constraints $Mu_i = v_i$ in the matrix form $MU = V$ for $U = (u_1, \ldots, u_q)$ and $V = (v_1, \ldots, v_q)$. Provided U has full column rank q, the minimum of the strictly convex function $\|M\|_F^2$ subject to the constraints is attained by the choice $M = V(U^tU)^{-1}U^t$.*

Proof If we take the partial derivative of the Lagrangian

$$\mathcal{L} = \frac{1}{2}\|M\|_F^2 + \operatorname{tr}\left[\Lambda^t(MU - V)\right]$$

with respect to m_{ij} and equate it to 0, then we get the Lagrange multiplier equation

$$0 = m_{ij} + \sum_k \lambda_{ik}u_{jk}.$$

These can be collectively expressed in matrix notation as $\mathbf{0} = M + \Lambda U^t$. This equation and the constraint equation $MU = V$ uniquely determine the minimum of the objective function. Straightforward substitution shows that $M = V(U^tU)^{-1}U^t$ and $\Lambda = -V(U^tU)^{-1}$ constitute the solution. □

To apply the proposition in our proposed quasi-Newton scheme, we must invert the matrix $I - V(U^tU)^{-1}U^t$. Fortunately, we have the explicit inverse

$$[I - V(U^tU)^{-1}U^t]^{-1} = I + V[U^tU - U^tV]^{-1}U^t.$$

The reader can readily check this variant of the Sherman-Morrison formula (Lange 1999). It is noteworthy that the $q \times q$ matrix $U^tU - U^tV$ is trivial to invert for q small even when p is large.

With these results in hand, the Newton update (1) can be replaced by the quasi-Newton update

$$\begin{aligned} x^{n+1} &= x^n - [I - V(U^tU)^{-1}U^t]^{-1}[x^n - F(x^n)] \\ &= x^n - [I + V(U^tU - U^tV)^{-1}U^t][x^n - F(x^n)] \\ &= F(x^n) - V(U^tU - U^tV)^{-1}U^t[x^n - F(x^n)]. \end{aligned}$$

The special case $q = 1$ is interesting in its own right. In this case the secant ingredients are clearly $u = F(x^n) - x^n$ and $v = F \circ F(x^n) - F(x^n)$. A brief calculation lets us express the quasi-Newton update as

$$x^{n+1} = (1 - c^n)F(x^n) + c^n F \circ F(x^n), \qquad (2)$$

where

$$\begin{aligned} c^n &= -\frac{\|F(x^n) - x^n\|^2}{[F \circ F(x^n) - 2F(x^n) + x^n]^t[F(x^n) - x^n]} \\ &= -\frac{u^tu}{u^t(v - u)}. \end{aligned}$$

The acceleration (2) differs from the squared extrapolation acceleration proposed by Varadhan and Roland (2008). In their SQUAREM acceleration

$$\begin{aligned} x^{n+1} &= x^n - 2s[F(x^n) - x^n] \\ &\quad + s^2[F \circ F(x^n) - 2F(x^n) + x^n] \\ &= x^n - 2su + s^2(v - u), \end{aligned}$$

where s is a scalar steplength. The versions of SQUAREM diverge in how they compute s:

$$\text{SqS1:} \quad s = \frac{u^tu}{u^t(v - u)},$$

$$\text{SqS2:} \quad s = \frac{u^t(v - u)}{(v - u)^t(v - u)},$$

$$\text{SqS3:} \quad s = -\sqrt{\frac{u^tu}{(v - u)^t(v - u)}}.$$

We will compare the performance of quasi-Newton acceleration and SQUAREM in several concrete examples.

Thus, the quasi-Newton method is feasible for high-dimensional problems and potentially faster than SQUAREM if we take $q > 1$. It takes two ordinary iterates to generate a secant condition and quasi-Newton update. If the quasi-Newton update fails to send the objective function in the right direction, then with an ascent or descent algorithm one can always revert to the second iterate $F \circ F(x^n)$. For a given q, we propose doing q initial ordinary updates and forming $q - 1$ secant pairs. At that point, quasi-Newton updating can commence. After each accelerated update, we replace the earliest retained secant pair by the new secant pair.

On the basis of the evidence presented by Varadhan and Roland (2008), we assume that SQUAREM is the current gold standard for acceleration. Hence, it is crucial to compare the behavior of quasi-Newton updates to SQUAREM on high-dimensional problems. There are good reasons for optimism. First, earlier experience (Lange 1995; Jamshidian and Jennrich 1997) with quasi-Newton methods was positive. Second, the effort per iteration is relatively light: two ordinary iterates and some matrix times vector multiplications. Most of the entries of U^tU and U^tV can be computed once and used over multiple iterations. Third, the whole acceleration scheme is consistent with linear constraints. Thus, if the parameter space satisfies a linear constraint $w^tx = a$ for all feasible x, then the quasi-Newton iterates also satisfy $w^tx^n = a$ for all n. This claim follows from the equalities $w^tF(x) = a$ and $w^tV = \mathbf{0}$ in the above notation. Finally, the recipe for constructing the approximation M to $dF(x^\infty)$ feels right, being the minimum M consistent with the secant conditions.

3 Examples

In this section, we compare the performance of the quasi-Newton acceleration and the SQUAREMs on various examples, including: (a) a truncated beta-binomial model, (b) a Poisson admixture model, (c) estimation of the location and scale for the multivariate t distribution, (d) an admixture problem in genetics, (e) PET imaging, (f) a movie rating problem, and (g) computation of the largest and smallest generalized eigenvalues of two large symmetric matrices. The number of parameters ranges from two to tens of thousands. For examples (a) and (b) with only a few parameters, whenever the accelerated step occurs outside the feasible region, we fall back to the most recent iterate of the original ascent or descent algorithm. For large scale problems (d), (e) and (f), we always project the accelerated point back to the feasible region. In most examples, we iterate until the relative change of the objective function between successive iterations falls below a pre-set threshold. In other words, we stop at iteration n when

$$\frac{|O^n - O^{n-1}|}{|O^{n-1}| + 1} \le \varepsilon,$$

where $\varepsilon > 0$ is small and O^n and O^{n-1} represent two successive values of the objective function under the unaccelerated algorithm. In most cases the objective function is a log-likelihood. We compare the performance of the different algorithms in terms of the number of evaluations of the algorithm map, the value of the objective function at termination, and running times. Computations for the genetics admixture and PET imaging problems were performed using the C++ programming language. All other examples were handled in MATLAB. Running times are recorded in seconds using the `tic/toc` functions of MATLAB.

3.1 Truncated beta-binomial

In many discrete probability models, only data with positive counts are observed. Counts that are 0 are missing. The likelihood function takes the form

$$L(\theta|x) = \prod_{i=1}^{m} \frac{g(x_i \mid \theta)}{1 - g(0 \mid \theta)},$$

where $g(x|\theta)$ is a standard discrete distribution with parameter vector θ. For household disease incidence data, a commonly used model is beta-binomial with density

$$g(x \mid t, \pi, \alpha) = \binom{t}{x} \frac{\prod_{j=0}^{x-1}(\pi + j\alpha) \prod_{k=0}^{t-x-1}(1 - \pi + k\alpha)}{\prod_{l=0}^{t-1}(1 + l\alpha)},$$

$$x = 0, 1, \ldots, t,$$

where the parameters $\pi \in (0, 1)$ and $\alpha > 0$ (Griffiths 1973). Given m independent observations $x_1, \ldots, x_m$ from a truncated beta-binomial model with possibly variable batch sizes $t_1, \ldots, t_m$, an MM algorithm proceeds with updates

$$\alpha^{n+1} = \frac{\sum_{k=0}^{t-1} (\frac{s_{1k} k \alpha^n}{\pi^n + k\alpha^n} + \frac{s_{2k} k \alpha^n}{1 - \pi^n + k\alpha^n})}{\sum_{k=0}^{t-1} \frac{r_k k}{1 + k\alpha^n}},$$

$$\pi^{n+1} = \frac{\sum_{k=0}^{t-1} \frac{s_{1k} \pi^n}{\pi^n + k\alpha^n}}{\sum_{k=0}^{t-1} [\frac{s_{1k} \pi^n}{\pi^n + k\alpha^n} + \frac{s_{2k}(1 - \pi^n)}{1 - \pi^n + k\alpha^n}]},$$

where s_{1k}, s_{2k}, and r_k are the pseudo-counts

$$s_{1k} = \sum_{i=1}^{m} 1_{\{x_i \ge k+1\}},$$

$$s_{2k} = \sum_{i=1}^{m} \left[1_{\{x_i \le t-k-1\}} + \frac{g(0 \mid t_i, \pi^n, \alpha^n)}{1 - g(0 \mid t_i, \pi^n, \alpha^n)} \right],$$

$$r_k = \sum_{i=1}^{m} \left[1 + \frac{g(0 \mid t_i, \pi^n, \alpha^n)}{1 - g(0 \mid t_i, \pi^n, \alpha^n)} \right] 1_{\{t_i \ge k+1\}|}.$$

See Zhou and Lange (2009b) for a detailed derivation.

Table 1 The Lidwell and Somerville (1951) cold data on households of size 4 and corresponding MLEs under the truncated beta-binomial model

Household	Number of cases				MLE	
type	1	2	3	4	$\hat{\pi}$	$\hat{\alpha}$
(a)	15	5	2	2	0.0000	0.6151
(b)	12	6	7	6	0.1479	1.1593
(c)	10	9	2	7	0.0000	1.6499
(d)	26	15	3	9	0.0001	1.0594

Table 2 Comparison of algorithms for the Lidwell and Somerville Data. The starting point is $(\pi^0, \alpha^0) = (0.5, 1)$, the stopping criterion is $\varepsilon = 10^{-9}$, and the number of parameters is two

Data	Algorithm	$\ln L$	Evals	Time
(a)	MM	−25.2277	30209	10.5100
	$q = 1$	−25.2270	157	0.1164
	$q = 2$	−25.2276	36	0.0603
	SqS1	−25.2277	1811	0.8046
	SqS2	−25.2276	53	0.0589
	SqS3	−25.2275	39	0.0569
(b)	MM	−41.7286	2116	0.7872
	$q = 1$	−41.7286	423	0.2390
	$q = 2$	−41.7286	20	0.0526
	SqS1	−41.7286	165	0.1095
	SqS2	−41.7286	193	0.1218
	SqS3	−41.7286	111	0.0805
(c)	MM	−37.3592	25440	9.2434
	$q = 1$	−37.3582	787	0.4008
	$q = 2$	−37.3586	26	0.0573
	SqS1	−37.3590	3373	1.4863
	SqS2	−37.3588	2549	1.1283
	SqS3	−37.3591	547	0.2791
(d)	MM	−65.0421	28332	10.1731
	$q = 1$	−65.0402	1297	0.6255
	$q = 2$	−65.0410	24	0.0603
	SqS1	−65.0418	3219	1.4537
	SqS2	−65.0412	4327	1.9389
	SqS3	−65.0419	45	0.0621

As a numerical example, we revisit the cold incidence data of Lidwell and Somerville (1951) summarized in Table 1. Zero-truncated models apply here because only households with at least one affected person are reported. The households were classified as: (a) adults only; (b) adults and school children; (c) adults and infants; and (d) adults, school children, and infants. Table 2 lists the number of MM evaluations, final log-likelihood, and running times until convergence for each acceleration tested. The starting

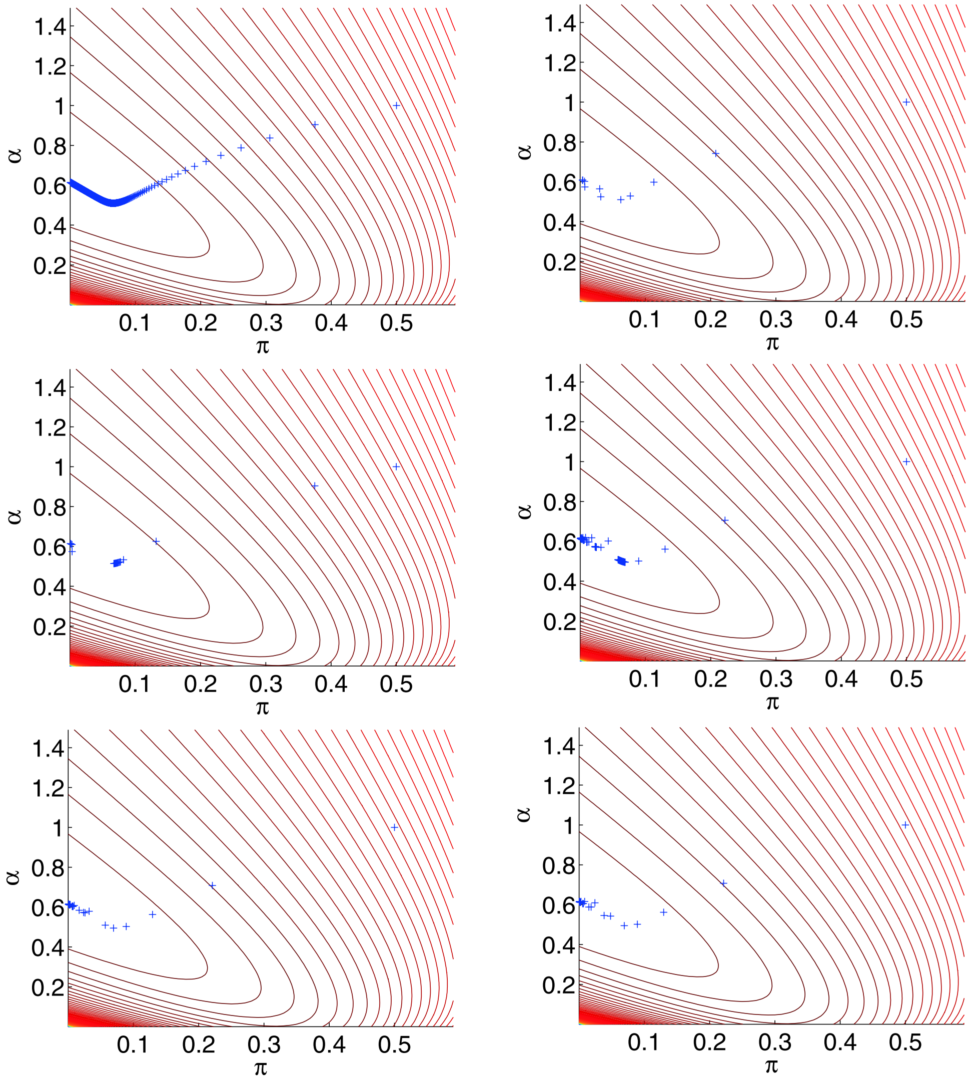

Fig. 1 Ascent of the different algorithms for the Lidwell and Somerville household type (a) data starting from $(\pi^0, \alpha^0) = (0.5, 1)$ with stopping criterion $\varepsilon = 10^{-9}$. *Top left*: naive MM; *Top right*: $q = 1$; *Middle left*: $q = 2$; *Middle right*: SqS1; *Bottom left*: SqS2; *Bottom right*: SqS3

point is $(\pi^0, \alpha^0) = (0.5, 1)$, and the stopping criterion is $\varepsilon = 10^{-9}$. Convergence is excruciatingly slow under the MM algorithm. Both the quasi-Newton acceleration and SQUAREM methods significantly reduce the number of iterations and time until convergence. Figure 1 depicts the progress of the different algorithms for the household type (a) data. Note the giant leaps made by the accelerated algorithms. For all four data sets, the quasi-Newton acceleration with $q = 2$ shows the best performance, consistently cutting time to convergence by two to three orders of magnitude.

3.2 Poisson admixture model

Consider the mortality data from *The London Times* (Titterington et al. 1985) during the years 1910–1912 presented in Table 3. The table alternates two columns giving the number of deaths to women 80 years and older reported by day and the number of days with i deaths. A Poisson distribution gives a poor fit to these frequency data, possibly because of different patterns of deaths in winter and summer. A mixture of two Poissons provides a much better fit. Under the Poisson admixture model, the likelihood of the observed data is

$$\prod_{i=0}^{9}\left[\pi e^{-\mu_1}\frac{\mu_1^i}{i!} + (1-\pi)e^{-\mu_2}\frac{\mu_2^i}{i!}\right]^{n_i},$$

where π is the admixture parameter and μ_1 and μ_2 are the means of the two Poisson distributions. The standard EM updates are

$$\mu_1^{n+1} = \frac{\sum_i n_i i w_i^n}{\sum_i n_i w_i^n}, \qquad \mu_2^{n+1} = \frac{\sum_i n_i i[1 - w_i^n]}{\sum_i n_i[1 - w_i^n]},$$

$$\pi^{n+1} = \frac{\sum_i n_i w_i^n}{\sum_i n_i},$$

Table 3 Death notices from *The London Times*

Deaths i	Frequency n_i	Death i	Frequency n_i
0	162	5	61
1	267	6	27
2	271	7	8
3	185	8	3
4	111	9	1

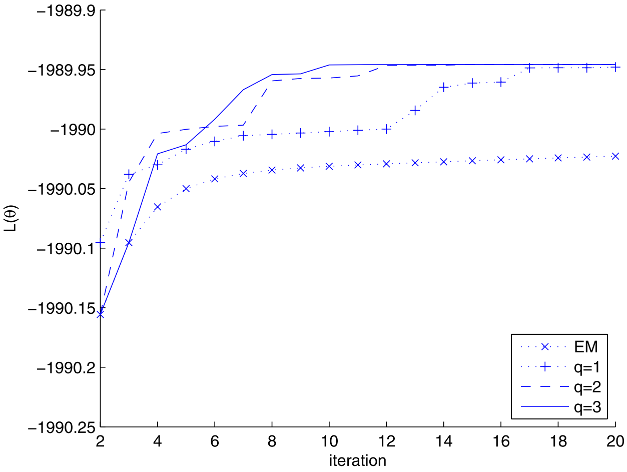

Fig. 2 EM acceleration for the Poisson admixture example

where w_i^n are the weights

$$w_i^n = \frac{\pi^n e^{-\mu_1^n}(\mu_1^n)^i}{\pi^n e^{-\mu_1^n}(\mu_1^n)^i + \pi^n e^{-\mu_2^n}(\mu_2^n)^i}.$$

The original EM algorithm converges slowly. Starting from the method of moment estimates

$$(\mu_1^0, \mu_2^0, \pi^0) = (1.101, 2.582, .2870),$$

it takes 652 iterations for the log-likelihood $L(\theta)$ to attain its maximum value of -1989.9461 and 1749 iterations for the parameters to reach the maximum likelihood estimates $(\hat{\mu}_1, \hat{\mu}_2, \hat{\pi}) = (1.256, 2.663, .3599)$. Despite providing a better fit, the three parameter likelihood surface is very flat. In contrast, the quasi-Newton accelerated EM algorithm converges to the maximum likelihood in only a few dozen iterations, depending on the choice of q. Figure 2 shows the progress of the various algorithms over the first 20 iterations. Table 4 compares their EM algorithm map evaluations, final log-likelihood, and running times. Here the quasi-Newton acceleration with $q = 3$ performs best, showing a 40-fold decrease in the number of EM map evaluations compared to the unaccelerated EM algorithm.

Table 4 Comparison of different algorithms on the Poisson admixture model. The starting point is $(\pi^0, \mu_1^0, \mu_2^0) = (0.2870, 1.101, 2.582)$, the stopping criterion is $\varepsilon = 10^{-9}$, and the number of parameters is three

Algorithm	$\ln L$	Evals	Time
EM	−1989.9461	652	0.0422
$q = 1$	−1989.9460	27	0.0031
$q = 2$	−1989.9459	38	0.0037
$q = 3$	−1989.9459	15	0.0025
SqS1	−1989.9459	41	0.0052
SqS2	−1989.9461	257	0.0294
SqS3	−1989.9459	31	0.0054

3.3 Multivariate t distribution

The multivariate t distribution is often employed as a robust substitute for the normal distribution in data fitting (Lange et al. 1989). For location vector $\mu \in \mathbb{R}^p$, positive definite scale matrix $\Omega \in \mathbb{R}^{p\times p}$, and degrees of freedom $\alpha > 0$, the multivariate t distribution has density

$$\frac{\Gamma(\frac{\alpha+p}{2})}{\Gamma(\frac{\alpha}{2})(\alpha\pi)^{p/2}|\Omega|^{1/2}[1+\frac{1}{\alpha}(x-\mu)^t\Omega^{-1}(x-\mu)]^{(\alpha+p)/2}},$$

for $x \in \mathbb{R}^p$. The standard EM updates (Lange et al. 1989) are

$$\mu^{n+1} = \frac{1}{v^n}\sum_{i=1}^m w_i^n x_i,$$

$$\Omega^{n+1} = \frac{1}{m}\sum_{i=1}^m w_i^n (x_i - \mu^{n+1})(x_i - \mu^{n+1})^t,$$

where $v^n = \sum_{i=1}^m w_i^n$ is the sum of the case weights

$$w_i^n = \frac{\alpha + p}{\alpha + d_i^n}, \quad d_i^n = (x_i - \mu^n)^t(\Omega^n)^{-1}(x_i - \mu^n).$$

An alternative faster algorithm (Kent et al. 1994; Meng and van Dyk 1997) updates Ω by

$$\Omega^{n+1} = \frac{1}{v^n}\sum_{i=1}^m w_i^n (x_i - \mu^{n+1})(x_i - \mu^{n+1})^t.$$

This faster version is called the parameter expanded EM (PX-EM) algorithm.

Table 5 reports the performances of the different algorithms on 100 simulated data sets each with 100 replicates of a 10-variate t distribution with 0.5 degrees of freedom. We used the original EM, PX-EM, quasi-Newton acceleration with $q = 0, \ldots, 5$, and SQUAREM algorithms to estimate μ and Ω at fixed degrees of freedom 1, 0.5, 0.1, and 0.05. The sample median vector and covariance matrix served as initial values for μ and Ω. The quasi-Newton accelerations of

Table 5 Comparison of the various algorithms for estimating the location and scale of a 10-variate t distribution with 0.5 degrees of freedom. The column $\ln L$ lists the average converged log-likelihood, and the column Evals lists the average number of EM evaluations. Running times are averaged over 100 simulations with 200 sample points each. The number of parameters is 65, and the stopping criterion is 10^{-9}

D.F.	Method	EM			PX-EM		
		$\ln L$	Evals	Time	$\ln L$	Evals	Time
1	EM	−3981.5470	160	0.8272	−3981.5470	15	0.0771
	$q = 1$	−3981.5470	26	0.1363	−3981.5470	10	0.0497
	$q = 2$	−3981.5470	22	0.1184	−3981.5470	10	0.0510
	$q = 3$	−3981.5470	23	0.1216	−3981.5470	10	0.0540
	$q = 4$	−3981.5470	24	0.1282	−3981.5470	11	0.0555
	$q = 5$	−3981.5470	26	0.1381	−3981.5470	11	0.0558
	SqS1	−3981.5470	29	0.1570	−3981.5470	10	0.0509
	SqS2	−3981.5470	31	0.1646	−3981.5470	10	0.0507
	SqS3	−3981.5470	30	0.1588	−3981.5470	10	0.0507
0.5	EM	−3975.8332	259	1.3231	−3975.8332	15	0.0763
	$q = 1$	−3975.8332	31	0.1641	−3975.8332	10	0.0506
	$q = 2$	−3975.8332	25	0.1343	−3975.8332	10	0.0512
	$q = 3$	−3975.8332	27	0.1405	−3975.8332	10	0.0544
	$q = 4$	−3975.8332	28	0.1479	−3975.8332	10	0.0547
	$q = 5$	−3975.8332	30	0.1553	−3975.8332	11	0.0552
	SqS1	−3975.8332	34	0.1829	−3975.8332	10	0.0514
	SqS2	−3975.8332	38	0.2017	−3975.8332	10	0.0513
	SqS3	−3975.8332	35	0.1895	−3975.8332	10	0.0513
0.1	EM	−4114.2561	899	4.5996	−4114.2561	16	0.0816
	$q = 1$	−4114.2562	52	0.2709	−4114.2561	10	0.0521
	$q = 2$	−4114.2561	36	0.1924	−4114.2561	10	0.0533
	$q = 3$	−4114.2561	34	0.1820	−4114.2561	10	0.0544
	$q = 4$	−4114.2561	36	0.1895	−4114.2561	10	0.0544
	$q = 5$	−4114.2561	38	0.2041	−4114.2561	11	0.0558
	SqS1	−4114.2561	51	0.2717	−4114.2561	10	0.0522
	SqS2	−4114.2561	66	0.3492	−4114.2561	10	0.0518
	SqS3	−4114.2561	54	0.2846	−4114.2561	10	0.0519
0.05	EM	−4224.9190	1596	8.1335	−4224.9190	17	0.0857
	$q = 1$	−4224.9192	62	0.3248	−4224.9190	10	0.0530
	$q = 2$	−4224.9192	47	0.2459	−4224.9190	10	0.0539
	$q = 3$	−4224.9191	39	0.2006	−4224.9190	10	0.0549
	$q = 4$	−4224.9191	40	0.2089	−4224.9190	11	0.0564
	$q = 5$	−4224.9191	42	0.2239	−4224.9190	11	0.0565
	SqS1	−4224.9191	60	0.3156	−4224.9190	10	0.0543
	SqS2	−4224.9191	91	0.4809	−4224.9190	10	0.0535
	SqS3	−4224.9191	64	0.3417	−4224.9190	10	0.0535

the EM algorithm with $q > 1$ outperform the SQUAREM algorithms. For the PX-EM algorithm, there is not much room for improvement.

3.4 A genetic admixture problem

A genetic admixture problem described in Alexander et al. (2009) also benefits from quasi-Newton acceleration. Modern genome-wide association studies type a large sample of unrelated individuals at many SNP (single nucleotide polymorphism) markers. As a prelude to the mapping of disease genes, it is a good idea to account for hidden population stratification. The problem thus becomes one of estimating the ancestry proportion of each sample individual attributable to each of K postulated founder populations. The unknown parameters are the allele frequencies $F = \{f_{kj}\}$ for the J markers and K populations and the admixture coefficients $W = \{w_{ik}\}$ for the I sample people. The admixture coefficient w_{ik} is loosely defined as the probability that a random gene taken from individual i originates from population k; these proportions obey the constraint $\sum_{k=1}^{K} w_{ik} = 1$. Under the assumption that individual i's genotype at SNP j is formed by random sampling of gametes, we have

$$\Pr(\text{genotype } 1/1 \text{ for } i \text{ at SNP } j) = \Big[\sum_k w_{ik} f_{kj}\Big]^2,$$

$$\Pr(\text{genotype } 1/2 \text{ for } i \text{ at SNP } j) = 2\Big[\sum_k w_{ik} f_{kj}\Big]\Big[\sum_k w_{ik}(1-f_{kj})\Big],$$

$$\Pr(\text{genotype } 2/2 \text{ for } i \text{ at SNP } j) = \Big[\sum_k w_{ik}(1-f_{kj})\Big]^2.$$

Note here that the SNP has two alleles labeled 1 and 2. Under the further assumptions that the SNPs are in linkage equilibrium, we can write the log-likelihood for the entire dataset as

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\Big\{n_{ij}\ln\Big[\sum_k w_{ik} f_{kj}\Big] + (2-n_{ij})\ln\Big[\sum_k w_{ik}(1-f_{kj})\Big]\Big\},$$

where n_{ij} is the number of alleles of type 1 individual i possesses at SNP j.

In estimating the parameters by maximum likelihood, Newton's method and scoring are out of the question because they require storing and inverting a very large information matrix. It is easy to devise an EM algorithm for this problem, but its performance is poor because many parameters wind up on boundaries. We have had greater success with a block relaxation algorithm that alternates updates of the W and F parameter matrices. Block relaxation generates a smooth algorithm map and involves only small decoupled optimizations. These are relatively straightforward to solve using sequential quadratic programming. As with MM, block relaxation enjoys the desirable ascent property.

We implemented block relaxation with acceleration on a sample of 912 European American controls genotyped at 9,378 SNPs as part of an inflammatory bowel disease (IBD) study (Mitchell et al. 2004). The data show strong evidence of northwestern European and Ashkenazi Jewish ancestry. The evidence for southeastern European ancestry is less compelling. With $K = 3$ ancestral populations there are 30,870 parameters to estimate. Although block relaxation converges, it can be substantially accelerated by both quasi-Newton and SQUAREM extrapolation methods.

Table 6 Comparison of acceleration algorithms for the genetics admixture problem with $K = 3$ ancestral populations using an IBD dataset (Mitchell et al. 2004) with 912 individuals and 9,378 SNP markers. The number of parameters is 30,870

Algorithm	Evals	$\ln L$	Time
Block relax.	169	−9183720.22	1055.61
$q = 1$	32	−9183720.21	232.02
$q = 2$	44	−9183720.21	346.84
$q = 3$	36	−9183720.21	276.47
$q = 4$	36	−9183720.21	260.33
$q = 5$	32	−9183720.21	225.01
$q = 6$	32	−9183720.21	212.09
$q = 7$	34	−9183720.21	224.71
$q = 8$	38	−9183720.21	251.59
$q = 9$	36	−9183720.21	232.29
$q = 10$	44	−9183720.21	291.80
$q = 15$	46	−9183720.21	289.10
$q = 20$	54	−9183720.21	339.72
SqS1	32	−9183720.21	230.35
SqS2	38	−9183720.21	276.29
SqS3	30	−9183720.21	214.83

In this example, the best-performing acceleration method is SqS3, with a 5.6-fold reduction in the number of block-relaxation algorithm evaluations. SqS3 narrowly edges out the best quasi-Newton accelerations ($q = 1$, 5, and 6).

3.5 PET imaging

The EM algorithm has been exploited for many years in the field of computed tomography. Acceleration of the classic algorithm (Lange and Carson 1984; Vardi et al. 1985) for PET imaging (positron emission tomography) was explored by Varadhan and Roland (2004). The problem consists of estimating Poisson emission intensities $\lambda = (\lambda_1, \ldots, \lambda_p)$ for p pixels arranged in a 2-dimensional grid and surrounded by photon detectors. The observed data are coincidence counts $(y_1, y_2, \ldots, y_d)$ along d lines of flight connecting pairs of photon detectors. The observed and complete data log-likelihoods for the PET model are

$$L_{\text{observed}}(\lambda) = \sum_i \Big[y_i \ln\Big(\sum_j c_{ij}\lambda_j\Big) - \sum_j c_{ij}\lambda_j\Big],$$

$$L_{\text{complete}}(\lambda) = \sum_i \sum_j [z_{ij}\log(\lambda_j c_{ij}) - \lambda_j c_{ij}],$$

where the c_{ij} are constants derived from the geometry of the grid and the detectors, and the missing data variable z_{ij} counts the number of emission events emanating from pixel j directed along line of flight i. Without loss of generality, one can assume $\sum_i c_{ij} = 1$ for each j.

The E step of the EM algorithm replaces z_{ij} by its conditional expectation

$$z_{ij}^n = \frac{y_i c_{ij} \lambda_j^n}{\sum_k c_{ik} \lambda_k^n}$$

given the data y_i and the current parameter values λ_j^n. Maximization of $L_{\text{complete}}(\lambda)$ with this substitution yields the EM updates

$$\lambda_j^{n+1} = \sum_i z_{ij}^n.$$

Full details can be found in McLachlan and Krishnan (2008).

In experimenting with this EM algorithm, we found convergence to the maximum likelihood estimate to be frustratingly slow, even under acceleration. Furthermore, maximum likelihood yielded a reconstructed image of poor quality with a grainy appearance. The traditional remedy of premature halting of the algorithm cuts computational cost but does not lend itself well to comparing different methods of acceleration. A better option is to add a roughness penalty to the observed log-likelihood. This device has long been known to produce better images and accelerate convergence. Thus, we maximize the amended objective function

$$f(\lambda) = L_{\text{observed}}(\lambda) - \frac{\mu}{2} \sum_{\{j,k\} \in N} (\lambda_j - \lambda_k)^2,$$

where μ is the roughness penalty constant, and N is the neighborhood system. A pixel pair $\{j,k\} \in N$ if and only if j and k are spatially adjacent. Although an absolute value penalty is less likely to deter the formation of edges than a square penalty, it is easier to deal with a square penalty analytically, and we adopt it for the sake of simplicity. In practice, the roughness penalty μ can be chosen by visual inspection of the recovered images.

To maximize $f(\lambda)$ by an MM algorithm, we first minorize the log-likelihood part of $f(\lambda)$ by the surrogate function

$$Q(\lambda \mid \lambda^n) = \sum_i \sum_j [z_{ij}^n \log(\lambda_j c_{ij}) - \lambda_j c_{ij}]$$

derived from the E step of the EM algorithm. Here we have omitted an irrelevant constant that does not depend on the current parameter vector λ. To minorize the penalty, we capitalize on the evenness and convexity of the function x^2. Application of these properties yields the inequality

$$(\lambda_j - \lambda_k)^2 \le \frac{1}{2}(2\lambda_j - \lambda_j^n - \lambda_k^n)^2 + \frac{1}{2}(2\lambda_k - \lambda_j^n - \lambda_k^n)^2.$$

Equality holds for $\lambda_j + \lambda_k = \lambda_j^n + \lambda_k^n$, which is true when $\lambda = \lambda^n$. Combining our two minorizations gives the surrogate function

$$\begin{aligned} g(\lambda \mid \lambda^n) \\ &= Q(\lambda \mid \lambda^n) \\ &\quad - \frac{\mu}{4} \sum_{\{j,k\} \in N} \left[(2\lambda_j - \lambda_j^n - \lambda_k^n)^2 + (2\lambda_k - \lambda_j^n - \lambda_k^n)^2\right]. \end{aligned}$$

In maximizing $g(\lambda \mid \lambda^n)$, we set the partial derivative

$$\frac{\partial g}{\partial \lambda_j} = \sum_i \left[\frac{z_{ij}^n}{\lambda_j} - c_{ij}\right] - \mu \sum_{k \in N_j} (2\lambda_j - \lambda_j^n - \lambda_k^n) \tag{3}$$

equal to 0 and solve for λ^{n+1}. Here N_j is the set of pixels k with $\{j,k\} \in N$. Multiplying equation (3) by λ_j produces a quadratic with roots of opposite signs; we take the positive root as λ_j^{n+1}. If we set $\mu = 0$, then we recover the pure-EM solution.

Results from running the various algorithms on a simulated dataset (kindly provided by Ravi Varadhan) with 4,096 parameters (pixels) and observations from 2,016 detectors are shown in Table 7. In all cases, we took the roughness-penalty constant to be $\mu = 10^{-6}$ and the convergence criterion to be $\varepsilon = 10^{-8}$. Here, the best performing quasi-Newton methods ($q = 6$ through 10 and 15) edge out SqS3, the best of the SQUAREM methods.

Table 7 Comparison of various algorithms for the PET imaging problem. A 4,096-pixel image is recovered from photon coincidence counts collected from 2,016 detector tubes. Here the roughness constant is $\mu = 10^{-6}$, and the convergence criterion is $\varepsilon = 10^{-8}$. The number of parameters is 4,096

Algorithm	Evals	Objective	Time
EM	3376	−15432.61	6836.05
$q = 1$	740	−15432.61	1743.55
$q = 2$	608	−15432.60	1432.38
$q = 3$	406	−15432.60	955.30
$q = 4$	372	−15432.57	875.51
$q = 5$	268	−15432.58	627.00
$q = 6$	222	−15432.57	520.97
$q = 7$	204	−15432.56	477.02
$q = 8$	188	−15432.54	441.48
$q = 9$	178	−15432.52	417.63
$q = 10$	176	−15432.51	411.20
$q = 15$	184	−15432.62	430.94
$q = 20$	236	−15432.45	559.32
SqS1	314	−15435.72	742.90
SqS2	290	−15432.54	684.79
SqS3	232	−15432.53	549.06

3.6 Movie rating

In this example, we accelerate an EM algorithm for movie rating (Zhou and Lange 2009a). Suppose a website or company asks consumers to rate movies on an integer scale from 1 to d; typically $d = 5$ or 10. Let M_i be the set of movies rated by person i. Denote the cardinality of M_i by $|M_i|$. Each rater does so in one of two modes that we will call "quirky" and "consensus". In quirky mode rater i has a private rating distribution with discrete density $q(x \mid \alpha_i)$ that applies to every movie regardless of its intrinsic merit. In consensus mode, rater i rates movie j according to a discrete density $c(x \mid \beta_j)$ shared with all other raters in consensus mode. For every movie i rates, he or she makes a quirky decision with probability π_i and a consensus decision with probability $1 - \pi_i$. These decisions are made independently across raters and movies. If x_{ij} is the rating given to movie j by rater i, then the likelihood of the data is

$$L(\theta) = \prod_i \prod_{j \in M_i} \left[\pi_i q(x_{ij} \mid \alpha_i) + (1 - \pi_i) c(x_{ij} \mid \beta_j)\right], \qquad (4)$$

where $\theta = (\pi, \alpha, \beta)$ is the parameter vector of the model. Once we estimate the parameters, we can rank the reliability of rater i by the estimate $\hat{\pi}_i$ and the popularity of movie j by its estimated average rating $\sum_k kc(k \mid \hat{\beta}_j)$ in consensus mode.

Among the many possibilities for the discrete densities $q(x \mid \alpha_i)$ and $c(x \mid \beta_j)$, we confine ourselves to the shifted binomial distribution with $d - 1$ trials and values $1, \ldots, d$ rather than $0, \ldots, d - 1$. The discrete densities are

$$q(k \mid \alpha_i) = \binom{d-1}{k-1} \alpha_i^{k-1} (1 - \alpha_i)^{d-k},$$

$$c(k \mid \beta_j) = \binom{d-1}{k-1} \beta_j^{k-1} (1 - \beta_j)^{d-k},$$

where the binomial parameters α_i and β_j occur on the unit interval $[0, 1]$. The EM updates

$$\pi_i^{n+1} = \frac{1}{|M_i|} \sum_{j \in M_i} w_{ij}^n,$$

$$\alpha_i^{n+1} = \frac{\sum_{j \in M_i} w_{ij}^n (x_{ij} - 1)}{(d-1) \sum_{j \in M_i} w_{ij}^n},$$

$$\beta_j^{n+1} = \frac{\sum_{i: j \in M_i} (1 - w_{ij}^n)(x_{ij} - 1)}{(d-1) \sum_{i: j \in M_i} (1 - w_{ij}^n)}$$

are easy to derive (Zhou and Lange 2009a). Here the weights

$$w_{ij}^n = \frac{\pi_i^n q(x_{ij} \mid \alpha_i^n)}{\pi_i^n q(x_{ij} \mid \alpha_i^n) + (1 - \pi_i^n) c(x_{ij} \mid \beta_j^n)}$$

reflect Bayes' rule. The boundaries 0 and 1 are sticky in the sense that a parameter started at 0 or 1 is trapped there forever. Hence, acceleration must be monitored. If an accelerated point falls on a boundary or exterior to the feasible region, then it should be projected to an interior point close to the boundary. Even with this modification, the algorithm can converge to an inferior mode.

Table 8 Comparison of accelerations for the movie rating problem. Here the starting point is $\pi_i = \alpha_i = \beta_j = 0.5$, the stopping criterion is $\varepsilon = 10^{-9}$, and the number of parameters equals 2,771

Algorithm	$\ln L$	Evals	Time
EM	−119085.2039	671	189.3020
$q = 1$	−119085.2020	215	64.1149
$q = 2$	−119085.1983	116	36.6745
$q = 3$	−119085.1978	153	46.0387
$q = 4$	−119085.1961	156	46.9827
$q = 5$	−119085.1974	161	48.6629
SqS1	−119085.2029	341	127.9918
SqS2	−119085.2019	301	110.9871
SqS3	−119085.2001	157	56.7568

We consider a representative data set sampled by the GroupLens Research Project at the University of Minnesota (movielens.umn.edu) during the seven-month period from September 19, 1997 through April 22, 1998. The data set consists of 100,000 movie ratings on a scale of 1 to 5 collected from 943 users on 1682 movies. To avoid sparse data, we discard movies or raters with fewer than 20 ratings. This leaves 94,443 ratings from 917 raters on 937 movies. If there are a raters and b movies, the shifted binomial model involves $2a + b$ free parameters. For the current data set, this translates to 2,771 free parameters. Table 8 summarizes the performance of the different accelerations. The quasi-Newton acceleration with $q = 2$ performs best, reducing the number of EM algorithm map evaluations by 5.8-fold.

3.7 Generalized eigenvalues

Given two $m \times m$ matrices A and B, the generalized eigenvalue problem consists of finding all scalars λ and corresponding nontrivial vectors x satisfying $Ax = \lambda Bx$. In the special case where A is symmetric and B is symmetric and positive definite, all generalized eigenvalues λ and generalized eigenvectors x are real. The preferred algorithm for solving the symmetric-definite generalized eigenvalue problem combines a Cholesky decomposition and a symmetric QR algorithm (Golub and Van Loan 1996, Algorithm 8.7.1). The number of floating point operations required is on the order of $14m^3$. The alternative QZ algorithm (Golub and Van Loan 1996, Algorithm 7.7.3) requires about $30m^3$ floating point operations.

In statistical applications such as principal component analysis (Hotelling 1933; Jolliffe 1986), canonical correlation analysis (Hotelling 1936), and Fisher's discriminant analysis, only a few of the largest generalized eigenvalues are of interest. In this situation the standard algorithms represent overkill, particularly for large m. Numerical analysts have formulated efficient Krylov subspace methods for finding extremal generalized eigenvalues (Saad 1992). Here we describe an alternative algorithm which is easier to implement. The key to progress is to reformulate the problem as optimizing the Rayleigh quotient

$$R(x) = \frac{x^t Ax}{x^t Bx} \tag{5}$$

over the domain $x \neq \mathbf{0}$. Because the gradient of $R(x)$ is

$$\nabla R(x) = \frac{2}{x^t Bx}[Ax - R(x)Bx],$$

a stationary point of $R(x)$ furnishes an eigen-pair. Maximizing $R(x)$ gives the largest eigenvalue, and minimizing $R(x)$ gives the smallest eigenvalue. Possible algorithms include steepest ascent and steepest descent. These are notoriously slow, but it is worth trying to accelerate them. Hestenes and Karush (1951a, 1951b) suggest performing steepest ascent and steepest descent with a line search.

Here are the details. Let x^n be the current iterate, and put $u = x^n$ and $v = [A - R(x^n)B]x^n$. We search along the line $c \mapsto u + cv$ emanating from u. There the Rayleigh quotient

$$R(u+cv) = \frac{(u+cv)^t A(u+cv)}{(u+cv)^t B(u+cv)}$$

reduces to a ratio of two quadratics in c. The coefficients of the powers of c for both quadratics can be evaluated by matrix-vector and inner product operations alone. No matrix-matrix operations are needed. The optimal points are found by setting the derivative

$$\begin{aligned} &2\big[(v^t Au + cv^t Av)(u+cv)^t B(u+cv) \\ &\quad - (u+cv)^t A(u+cv)(v^t Bu + cv^t Bv)\big] \\ &\quad \times [(u+cv)^t B(u+cv)]^{-2} \end{aligned}$$

with respect to c equal to 0 and solving for c. Conveniently, the coefficients of c^3 in the numerator of this rational function cancel. This leaves a quadratic that can be easily solved. One root gives steepest ascent, and the other root gives steepest descent. The sequence $R(x^n)$ usually converges to the requisite generalized eigenvalue. The analogous algorithm for the smallest eigenvalue is obvious.

Because of the zigzag nature of steepest ascent, naive acceleration performs poorly. If $x^{n+1} = F(x^n)$ is the algorithm map, we have found empirically that it is better to replace $F(x)$ by its s-fold functional composition $F_s(x)$ before attempting acceleration, where s is an even number. This substitution preserves the ascent property. Table 9 shows the results of accelerating two-step ($s = 2$) steepest ascent and steepest descent. Here we have averaged over 100 random trials with 100×100 symmetric matrices. The matrices A and B were generated as $A = C + C^t$ and $B = DD^t$, with the entries of both C and D chosen to be independent, identically distributed uniform deviates from the interval $[-5, 5]$. Every trial run commences with x^0 equal to the constant vector $\mathbf{1}$. In general, quasi-Newton acceleration improves as q increases. With $q = 10$, we see a more than 25-fold improvement in computational speed.

Table 9 Average number of $F(x)$ evaluations and running times for 100 simulated random matrices A and B of dimension 100×100. Here $s = 2$, the stopping criterion is $\varepsilon = 10^{-9}$, and the number of parameters is 100

Algorithm	Largest eigenvalue		Smallest eigenvalue	
	Evals	Time	Evals	Time
Naive	40785	7.5876	39550	7.2377
$q = 1$	8125	1.6142	8044	1.6472
$q = 2$	1521	0.3354	1488	0.3284
$q = 3$	1486	0.3302	1466	0.3384
$q = 4$	1435	0.3257	1492	0.3376
$q = 5$	1454	0.3250	1419	0.3305
$q = 6$	1440	0.3280	1391	0.3188
$q = 7$	1302	0.2959	1283	0.3041
$q = 8$	1298	0.3001	1227	0.2864
$q = 9$	1231	0.2838	1227	0.2931
$q = 10$	1150	0.2725	1201	0.2832
SqS1	5998	1.1895	6127	1.2538
SqS2	3186	0.6578	4073	0.8271
SqS3	2387	0.4922	3460	0.7246

4 Discussion

The EM algorithm is one of the most versatile tools in the statistician's toolbox. The MM algorithm generalizes the EM algorithm and shares its positive features. Among the assets of both algorithms are simplicity, stability, graceful adaptation to constraints, and the tendency to avoid large matrix inversions. Scoring and Newton's methods become less and less attractive as the number of parameters increases. Unfortunately, some EM and MM algorithms are notoriously slow to converge. This is cause for concern as statisticians head into an era dominated by large data sets and high-dimensional models. In order for the EM and MM algorithms to take up the slack left by competing algorithms, statisticians must find efficient acceleration schemes. The quasi-Newton scheme discussed in the current paper is one candidate.

Successful acceleration methods will be instrumental in attacking another nagging problem in computational statistics, namely multimodality. No one knows how often statistical inference is fatally flawed because a standard optimization algorithm converges to an inferior mode. The current remedy of choice is to start a search algorithm from multiple random points. Algorithm acceleration is welcome because the number of starting points can be enlarged without an increase in computing time. As an alternative, our recent paper (Zhou and Lange 2009c) suggests modifications of several standard MM algorithms that head reliably toward global maxima. These simple modifications all involve variations on deterministic annealing (Ueda and Nakano 1998).

Our acceleration scheme attempts to approximate Newton's method for finding a fixed point of the algorithm map. Like SQUAREM, our scheme is off-the-shelf and applies to any search method determined by a smooth algorithm map. The storage requirement is $O(mq)$, where m is number of parameters and q is number of secant conditions invoked. The effort per iteration is very light: two ordinary updates and some matrix times vector multiplications. The whole scheme is consistent with linear constraints. These properties make it attractive for modern high-dimensional problems. In our numerical examples, quasi-Newton acceleration performs similarly or better than SQUAREM. In defense of SQUAREM, it is a bit easier to code.

As mentioned in our introduction, quasi-Newton methods can be applied to either the objective function or the algorithm map. The objective function analog of our algorithm map procedure is called the limited-memory BFGS (LBFGS) update in the numerical analysis literature (Nocedal and Wright 2006). Although we have not done extensive testing, it is our impression that the two forms of acceleration perform comparably in terms of computational complexity and memory requirements. However, there are some advantages of working directly with the algorithm map. First, the algorithm map is often easier to code than the gradient of the objective function. Second, our map algorithm acceleration respects linear constraints. A systematic comparison of the two methods is worth pursuing. The earlier paper (Lange 1995) suggests an MM adaptation of LBFGS that preserves curvature information supplied by the surrogate function.

The current research raises as many questions as it answers. First, the optimal choice of the number of secant conditions q varied from problem to problem. Our examples suggest that high-dimensional problems benefit from larger q. However, this rule of thumb is hardly universal. Similar criticisms apply to SQUAREM, which exists in at least three different flavors. A second problem is that quasi-Newton acceleration may violate boundary conditions and nonlinear constraints. When the feasible region is intersection of a finite number of closed convex sets, Dykstra's algorithm (Sect. 11.3, Lange 2004) is handy in projecting wayward points back to the feasible region. Third, although quasi-Newton acceleration almost always boosts the convergence rate of an MM algorithm, there may be other algorithms that do even better. One should particularly keep in mind parameter expansion, block relaxation, or combinations of block relaxation with MM. The multivariate t example is a case in point. Neither the quasi-Newton nor the SQUAREM acceleration of the naive EM algorithm beats the PX-EM algorithm.

A final drawback of quasi-Newton acceleration is that it can violate the ascent or descent property of the original algorithm. This is a particular danger when accelerated points fall outside the feasible region and must be projected back to it. For the sake of simplicity in our examples, we revert to the original algorithm whenever the ascent or descent property fails. A more effective strategy might be back-tracking (Varadhan and Roland 2008). In back-tracking a bad step is contracted toward the default iterate. Contraction trades more evaluations of the objective function for faster overall convergence. It would be worth exploring these tradeoffs more carefully. Finally, in applications such as factor analysis, latent class analysis, and multidimensional scaling, the problems of multimodality and slow convergence are intermingled. This too is worthy of closer investigations. In the interests of brevity, we simply state these challenges rather than seriously address them. Even without resolving them, it seems to us that the overall quasi-Newton strategy has proved its worth.

Acknowledgements The authors thank the referees for their many valuable comments. K.L. was supported by United States Public Health Service grants GM53257 and MH59590.

References

Alexander, D.H., Novembre, J., Lange, K.L.: Fast model-based estimation of ancestry in unrelated individuals. Genome Res. **19**, 1655–1664 (2009)

Becker, M.P., Young, I., Lange, K.L.: EM algorithms without missing data. Stat. Methods Med. Res. **6**, 37–53 (1997)

de Leeuw, J.: Block relaxation algorithms in statistics. In: Bock, H.H., Lenski, W., Richter, M.M. (eds.) Information Systems and Data Analysis, pp. 308–325. Springer, New York (1994)

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

Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)

Griffiths, D.A.: Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of a disease. Biometrics **29**(4), 637–648 (1973)

Heiser, W.J.: Convergent computing by iterative majorization: theory and applications in multidimensional data analysis. In: Krzanowski, W.J. (ed.) Recent Advances in Descriptive Multivariate Analysis, pp. 157–189. Clarendon Press, Oxford (1995)

Hestenes, M.R., Karush, W.: A method of gradients for the calculation of the characteristic roots and vectors of a real symmetric matrix. J. Res. Natl. Bur. Stand. **47**, 45–61 (1951a)

Hestenes, M.R., Karush, W.: Solutions of $Ax = \lambda Bx$. J. Res. Natl. Bur. Stand. **47**, 471–478 (1951b)

Hotelling, H.: Analysis of a complex of statistical variables onto principal components. J. Educ. Psychol. **24**, 417–441 (1933)

Hotelling, H.: Relations between two sets of variables. Biometrika **28**, 321–377 (1936)

Hunter, D.R., Lange, K.L.: A tutorial on MM algorithms. Am. Stat. **58**, 30–37 (2004)

Jamshidian, M., Jennrich, R.I.: Conjugate gradient acceleration of the EM algorithm. J. Am. Stat. Assoc. **88**(421), 221–228 (1993)

Jamshidian, M., Jennrich, R.I.: Acceleration of the EM algorithm by using quasi-Newton methods. J. R. Stat. Soc. Ser. B **59**(3), 569–587 (1997)

Jolliffe, I.: Principal Component Analysis. Springer, New York (1986)

Kent, J.T., Tyler, D.E., Vardi, Y.: A curious likelihood identity for the multivariate t-distribution. Commun. Stat. Simul. Comput. **23**(2), 441–453 (1994)

Lange, K.L., Carson, R.: EM reconstruction algorithms for emission and transmission tomography. J. Comput. Assist. Tomogr. **8**(2), 306–316 (1984)

Lange, K.L.: A quasi-Newton acceleration of the EM algorithm. Stat. Sin. **5**(1), 1–18 (1995)

Lange, K.L.: Numerical Analysis for Statisticians. Springer, New York (1999)

Lange, K.L.: Optimization transfer using surrogate objective functions. J. Comput. Statist. **9**, 1–59 (2000)

Lange, K.L.: Optimization. Springer, New York (2004)

Lange, K.L., Little, R.J.A., Taylor, J.M.G.: Robust statistical modeling using the t distribution. J. Am. Stat. Assoc. **84**(408), 881–896 (1989)

Lidwell, O.M., Somerville, T.: Observations on the incidence and distribution of the common cold in a rural community during 1948 and 1949. J. Hyg. Camb. **49**, 365–381 (1951)

Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data, 2nd edn. Wiley-Interscience, New York (2002)

Liu, C., Rubin, D.B.: The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. Biometrika **81**(4), 633–648 (1994)

Liu, C., Rubin, D.B., Wu, Y.N.: Parameter expansion to accelerate EM: the PX-EM algorithm. Biometrika **85**(4), 755–770 (1998)

Louis, T.A.: Finding the observed information matrix when using the EM algorithm. J. R. Stat. Soc. Ser. B **44**(2), 226–233 (1982)

McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, 2nd edn. Wiley-Interscience, New York (2008)

Meng, X.L., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika **80**(2), 267–278 (1993)

Meng, X.L., van Dyk, D.: The EM algorithm—an old folk-song sung to a fast new tune (with discussion). J. R. Stat. Soc. Ser. B **59**(3), 511–567 (1997)

Mitchell, M., Gregersen, P., Johnson, S., Parsons, R., Vlahov, D.: The New York Cancer Project: rationale, organization, design, and baseline characteristics. J. Urban Health **61**, 301–310 (2004)

Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)

Saad, Y.: Numerical Methods for Large Eigenvalue Problems. Halstead [Wiley], New York (1992)

Titterington, D.M., Smith, A.F.M., Makov, U.E.: Statistical Analysis of Finite Mixture Distributions. Wiley, New York (1985)

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

Varadhan, R., Roland, C.: Squared extrapolation methods (squarem): a new class of simple and efficient numerical schemes for accelerating the convergence of the EM algorithm. Johns Hopkins University, Department of Biostatistics Working Papers (Paper 63) (2004)

Varadhan, R., Roland, C.: Simple and globally convergent methods for accelerating the convergence of any EM algorithm. Scand. J. Statist. **35**(2), 335–353 (2008)

Vardi, Y., Shepp, L.A., Kaufman, L.: A statistical model for positron emission tomography (with discussion). J. Am. Stat. Assoc. **80**(389), 8–37 (1985)

Wu, T.T., Lange, K.L.: The MM alternatives to EM. Stat. Sci. (2009, in press)

Zhou, H., Lange, K.L.: Rating movies and rating the raters who rate them. Am. Stat. **63**(4), 297–307 (2009)

Zhou, H., Lange, K.L.: MM algorithms for some discrete multivariate distributions. J. Comput. Graph. Stat. (2009b, to appear)

Zhou, H., Lange, K.L.: On the bumpy road to the dominant mode. Scand. J. Stat. (2009c, to appear)