International Statistical Review
Revue Internationale de Statistique

International Statistical Review (2014), 82, 1, 46–70 doi:10.1111/insr.12022

A Brief Survey of Modern Optimization for Statisticians[1]

Kenneth Lange[1,2], Eric C. Chi[2] and Hua Zhou[3]

[1]*Departments of Biomathematics and Statistics, University of California, Los Angeles, CA 90095-1766, USA*
E-mail: klange@ucla.edu
[2]*Department of Human Genetics, University of California, Los Angeles, CA 90095-1766, USA*
E-mail: ecchi@ucla.edu
[3]*Department of Statistics, North Carolina State University, Raleigh, NC 27695-8203, USA*
E-mail: hua_zhou@ncsu.edu

Summary

Modern computational statistics is turning more and more to high-dimensional optimization to handle the deluge of big data. Once a model is formulated, its parameters can be estimated by optimization. Because model parsimony is important, models routinely include non-differentiable penalty terms such as the lasso. This sober reality complicates minimization and maximization. Our broad survey stresses a few important principles in algorithm design. Rather than view these principles in isolation, it is more productive to mix and match them. A few well-chosen examples illustrate this point. Algorithm derivation is also emphasized, and theory is downplayed, particularly the abstractions of the convex calculus. Thus, our survey should be useful and accessible to a broad audience.

Key words: Block relaxation; Newton's method; MM algorithm; penalization; augmented Lagrangian; acceleration.

1 Introduction

Modern statistics represents a confluence of data, algorithms, practical inference, and subject area knowledge. As data mining expands, computational statistics is assuming greater prominence. Surprisingly, the confident prediction of the previous generation that Bayesian methods would ultimately supplant frequentist methods has given way to a realization that Markov chain Monte Carlo may be too slow to handle modern data sets. Size matters because large data sets stress computer storage and processing power to the breaking point. The most successful compromises between Bayesian and frequentist methods now rely on penalization and optimization. Penalties serve as priors and steer parameter estimates in realistic directions. In classical statistics, estimation usually meant least squares and maximum likelihood with smooth objective functions. In a search for sparse representations, mathematical scientists have introduced non-differentiable penalties such as the lasso and the nuclear norm. To survive in this alien terrain, statisticians are being forced to master exotic branches of mathematics such as convex calculus (Hiriart-Urruty & Lemarechal, 1996, 2001). Thus, the uneasy but productive relationship between statistics and mathematics continues, but in a different guise and mediated by new concerns.

[1] This paper is followed by discussions and a rejoinder.

The purpose of this survey article is to provide a few glimpses of the new optimization algorithms being crafted by computational statisticians and applied mathematicians. Although a survey of convex calculus for statisticians would certainly be helpful, our emphasis is more concrete. The truth of the matter is that a few broad categories of algorithms dominate. Furthermore, difficult problems require that several algorithmic pieces be assembled into a well-coordinated whole. Put another way, from a handful of basic ideas, computational statisticians often weave a complex tapestry of algorithms that meets the needs of a specific problem. No algorithm category should be dismissed a priori in tackling a new problem. There is plenty of room for creativity and experimentation. Algorithms are made for tinkering. When one part fails or falters, it can be replaced by a faster or more robust part.

This survey will treat the following methods: (a) block descent, (b) steepest descent, (c) Newton's method, quasi-Newton methods, and scoring, (d) the majorize–minimize (MM) and expectation–maximization (EM) algorithms, (e) penalized estimation, (f) the augmented Lagrangian method for constrained optimization, and (g) acceleration of fixed point algorithms. As we have mentioned, often the best algorithms combine several themes. We will illustrate the various themes by a sequence of examples. Although we avoid difficult theory and convergence proofs, we will try to point out along the way a few motivating ideas that stand behind most algorithms. For example, as its name indicates, steepest descent algorithms search along the direction of fastest decrease of the objective function. Newton's method and its variants all rely on the notion of local quadratic approximation, thus correcting the often poor linear approximation of steepest descent. In high dimensions, Newton's method stalls because it involves calculating and inverting large matrices of second derivatives.

The MM and EM algorithms replace the objective function by a simpler surrogate function. By design, optimizing the surrogate function sends the objective function downhill in minimization and uphill in maximization. In constructing the surrogate function for an EM algorithm, statisticians rely on notions of missing data. The more general MM algorithm calls on skills in inequalities and convex analysis. More often than not, concrete problems also involve parameter constraints. Modern penalty methods incorporate the constraints by imposing penalties on the objective function. A tuning parameter scales the strength of the penalties. In the classical penalty method, the constrained solution is recovered as the tuning parameter tends to infinity. In the augmented Lagrangian method, the constrained solution emerges for a finite value of the tuning parameter.

In the remaining sections, we adopt several notational conventions. Vectors and matrices appear in boldface type; for the most part, parameters appear as Greek letters. The differential $df(\boldsymbol{\theta})$ of a scalar-valued function $f(\boldsymbol{\theta})$ equals its row vector of partial derivatives; the transpose $\nabla f(\boldsymbol{\theta})$ of the differential is the gradient. The second differential $d^2 f(\boldsymbol{\theta})$ is the Hessian matrix of second partial derivatives. The Euclidean norm of a vector $\boldsymbol{b}$ and the spectral norm of a matrix $\boldsymbol{A}$ are denoted by $\|\boldsymbol{b}\|$ and $\|\boldsymbol{A}\|$, respectively. All other norms will be appropriately subscripted. The n-th entry b_n of a vector $\boldsymbol{b}$ must be distinguished from the n-th vector $\boldsymbol{b}_n$ in a sequence of vectors. To maintain consistency, b_{ni} denotes the i-th entry of $\boldsymbol{b}_n$. A similar convention holds for sequences of matrices.

2 Block Descent

Block relaxation (either block descent or block ascent) divides the parameters into disjoint blocks and cycles through the blocks, updating only those parameters within the pertinent block at each stage of a cycle (de Leeuw, 1994). For the sake of brevity, we consider only block descent. In updating a block, we minimize the objective function over the block. Hence, block

descent possesses the desirable descent property of always forcing the objective function downhill. When each block consists of a single parameter, block descent is called cyclic coordinate descent. The coordinate updates need not be explicit. In high-dimensional problems, implementation of one-dimensional Newton searches is often compatible with fast overall convergence. Block descent is best suited to unconstrained problems where the domain of the objective function reduces to a Cartesian product of the subdomains associated with the different blocks. Obviously, exact block updates are a huge advantage. Non-separable constraints can present insuperable barriers to coordinate descent because parameters get locked into place. In some problems, it is advantageous to consider overlapping blocks.

Example 1. *Non-negative least squares*

For a positive definite matrix $\boldsymbol{A} = (a_{ij})$ and vector $\boldsymbol{b} = (b_i)$, consider minimizing the quadratic function

$$f(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^t \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{b}^t\boldsymbol{\theta} + c$$

subject to the constraints $\theta_i \geq 0$ for all i. In the case of least squares, $\boldsymbol{A} = \boldsymbol{X}^t\boldsymbol{X}$ and $\boldsymbol{b} = -\boldsymbol{X}^t\boldsymbol{y}$ for some design matrix $\boldsymbol{X}$ and response vector $\boldsymbol{y}$. Equating the partial derivative of $f(\boldsymbol{\theta})$ with respect to θ_i to 0 gives

$$0 = \sum_j a_{ij}\theta_j + b_i.$$

Rearrangement now yields the unrestricted minimum

$$\theta_{n+1,i} = \theta_{ni} - \frac{1}{a_{ii}}\left(b_i + \sum_j a_{ij}\theta_{nj}\right).$$

Taking into account the non-negativity constraint, this must be amended to

$$\theta_{n+1,i} = \max\left\{0, \theta_{ni} - \frac{1}{a_{ii}}\left(b_i + \sum_j a_{ij}\theta_{nj}\right)\right\}$$

at stage $n+1$ to construct the coordinate descent update of θ_i.

Example 2. *Matrix factorization by alternating least squares*

In the 1960s, Kruskal (1965) applied the method of alternating least squares to factorial analysis of variance. Later, the subject was taken up by de Leeuw and colleagues (1990). Suppose $\boldsymbol{U}$ is a $m \times q$ matrix whose columns $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q$ represent data vectors. In many applications, it is reasonable to postulate a reduced number of prototypes $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$ and write

$$\boldsymbol{u}_j \approx \sum_{k=1}^{p} \boldsymbol{v}_k w_{kj}$$

for certain non-negative weights w_{kj}. The matrix $\boldsymbol{W} = (w_{kj})$ is $p \times q$. If p is small compared with q, then the representation $\boldsymbol{U} \approx \boldsymbol{V}\boldsymbol{W}$ compresses the data for easier storage and retrieval. Depending on the circumstances, one may want to add further constraints (Ding *et al.*, 2010).

For instance, if the entries of $\boldsymbol{U}$ are non-negative, then it is often reasonable to demand that the entries of $\boldsymbol{V}$ be non-negative as well (Lee & Seung, 1999; Paatero & Tapper, 1994). If we want each $\boldsymbol{u}_j$ to equal a convex combination of the prototypes, then constraining the column sums of $\boldsymbol{W}$ to equal 1 is indicated.

One way of estimating $\boldsymbol{V}$ and $\boldsymbol{W}$ is to minimize the squared Frobenius norm

$$\|\boldsymbol{U} - \boldsymbol{V}\boldsymbol{W}\|_{\mathrm{F}}^2 = \sum_{i=1}^{m}\sum_{j=1}^{q}\left(u_{ij} - \sum_{k=1}^{p} v_{ik}w_{kj}\right)^2.$$

No explicit solution is known, but alternating least squares offers an iterative attack. If $\boldsymbol{W}$ is fixed, then we can update the i-th row of $\boldsymbol{V}$ by minimizing the sum of squares

$$\sum_{j=1}^{q}\left(u_{ij} - \sum_{k=1}^{p} v_{ik}w_{kj}\right)^2.$$

Similarly, if $\boldsymbol{V}$ is fixed, then we can update the j-th column of $\boldsymbol{W}$ by minimizing the sum of squares

$$\sum_{i=1}^{n}\left(u_{ij} - \sum_{k=1}^{p} v_{ik}w_{kj}\right)^2.$$

Thus, block descent solves a sequence of least squares problems, some of which are constrained.

3 Steepest Descent

The first-order Taylor expansion

$$f(\boldsymbol{\theta} + \boldsymbol{\gamma}) = f(\boldsymbol{\theta}) + df(\boldsymbol{\theta})\boldsymbol{\gamma} + o(\|\boldsymbol{\gamma}\|)$$

of a differentiable function $f(\boldsymbol{\theta})$ around $\boldsymbol{\theta}$ motivates the method of steepest descent. In view of the Cauchy–Schwarz inequality, the choice

$$\boldsymbol{\gamma} = -\nabla f(\boldsymbol{\theta})/\|\nabla f(\boldsymbol{\theta})\|$$

minimizes the linear term $df(\boldsymbol{\theta})\boldsymbol{\gamma}$ of the expansion over the sphere of unit vectors. Of course, if $\nabla f(\boldsymbol{\theta}) = \boldsymbol{0}$, then $\boldsymbol{\theta}$ is a stationary point. The steepest descent algorithm iterates according to

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - s\nabla f(\boldsymbol{\theta}_n) \tag{1}$$

for some scalar $s > 0$. If s is sufficiently small, then the descent property $f(\boldsymbol{\theta}_{n+1}) < f(\boldsymbol{\theta}_n)$ holds. The most sophisticated version of the algorithm determines s by searching for the minimum of the objective function along the direction of steepest descent. Among the many methods of line search, the methods of false position, cubic interpolation, and golden section stand out (Lange, 2012). These are all local search methods, and unless some guarantee of convexity exists, confusion of local and global minima can occur.

The method of steepest descent often exhibits zigzagging and a painfully slow rate of convergence. For these reasons, it was largely replaced in practice by Newton's method and its variants. However, the sheer scale of modern optimization problems has led to a re-evaluation. The avoidance of second derivatives and Hessian approximations is now viewed as a virtue.

Furthermore, the method has been generalized to non-differentiable problems by substituting the forward directional derivative

$$d_{\boldsymbol{v}} f(\boldsymbol{\theta}) = \lim_{s \downarrow 0} \frac{f(\boldsymbol{\theta} + s\boldsymbol{v}) - f(\boldsymbol{\theta})}{s}$$

for the gradient (Tao *et al.*, 2010). Here, the idea is to choose a unit search vector $\boldsymbol{v}$ to minimize $d_{\boldsymbol{v}} f(\boldsymbol{\theta})$. In some instances, this secondary problem can be attacked by linear programming. For a convex problem, the condition $d_{\boldsymbol{v}} f(\boldsymbol{\theta}) \geq 0$ for all $\boldsymbol{v}$ is both necessary and sufficient for $\boldsymbol{\theta}$ to be a minimum point. If the domain of $f(\boldsymbol{\theta})$ equals a convex set C, then only tangent directions $\boldsymbol{v} = \boldsymbol{\mu} - \boldsymbol{\theta}$ with $\boldsymbol{\mu} \in C$ come into play.

Steepest descent also has a role to play in constrained optimization. Suppose we want to minimize $f(\boldsymbol{\theta})$ subject to the constraint $\boldsymbol{\theta} \in C$ for some closed convex set. The projected gradient method capitalizes on the steepest descent update (1) by projecting it onto the set C (Goldstein, 1964; Levitin & Polyak, 1966; Ruszczyński, 2006). It is well known that for a point $\boldsymbol{x}$ external to C, there is a closest point $P_C(\boldsymbol{x})$ to $\boldsymbol{x}$ in C. Explicit formulas for the projection operator $P_C(\boldsymbol{x})$ exist when C is a box, Euclidean ball, hyperplane, or half-space. Fast algorithms for computing $P_C(\boldsymbol{x})$ exist for the unit simplex, the ℓ_1 ball, and the cone of positive semidefinite matrices (Duchi *et al.*, 2008; Michelot, 1986).

Choice of the scalar s in the update (1) is crucial. Current theory suggests taking s to equal r/L, where L is a Lipschitz constant for the gradient $\nabla f(\theta)$ and r belongs to the interval $(0, 2)$. In particular, the Lipschitz inequality

$$\|\nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\gamma})\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\gamma}\|$$

is valid for $L = \sup_{\boldsymbol{\theta}} \|d^2 f(\boldsymbol{\theta})\|$, whenever this quantity is finite. In practice, the Lipschitz constant L must be estimated. Any induced matrix norm $\|\cdot\|_\dagger$ can be substituted for the spectral norm $\|\cdot\|$ in the defining supremum and will give an upper bound on L.

Example 3. *Coordinate descent versus the projected gradient method*

As a test problem, we generated a random 100×50 design matrix $\boldsymbol{X}$ with independent and identically distributed (i.i.d.) standard normal entries, a random 50×1 parameter vector $\boldsymbol{\theta}$ with i.i.d. uniform [0,1] entries, and a random 100×1 error vector $\boldsymbol{e}$ with i.i.d. standard normal entries. In this setting, the response $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{e}$. We then compared coordinate descent, the projected gradient method (for L equal to the spectral radius of $\boldsymbol{X}^t\boldsymbol{X}$ and r equal to 1.0, 1.75, and 2.0), and the MM algorithm explained later in Example 6. All computer runs start from the common point $\boldsymbol{\theta}_0$ whose entries are filled with i.i.d. uniform [0,1] random deviates. Figure 1 plots the progress of each algorithm as measured by the relative difference

$$\frac{f(\boldsymbol{\theta}_n) - f(\boldsymbol{\theta}_\infty)}{f(\boldsymbol{\theta}_\infty)}, \tag{2}$$

between the loss at the current iteration and the ultimate loss at convergence. It is interesting how well coordinate descent performs compared with projected gradient descent. The slower convergence of the MM algorithm is probably a consequence of the fact that its multiplicative updates slow down as they approach the 0 boundary. Note also the importance of choosing a good step size in the projected gradient algorithm. Inflated steps accelerate convergence, but excessively inflated steps hamper it.

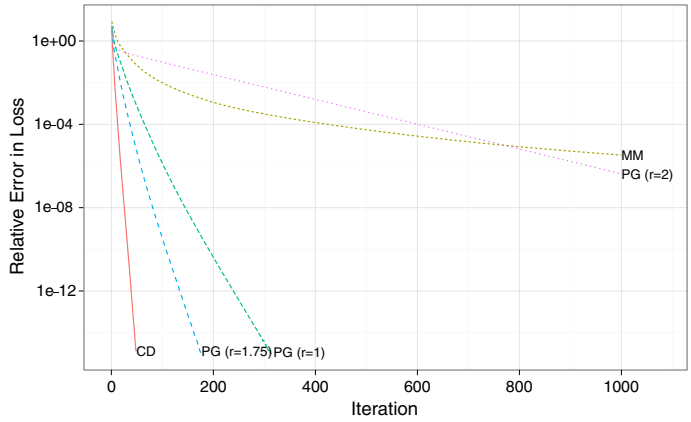

Figure 1. *Comparing the rate of convergence of three algorithms on a non-negative least squares problem. CD, coordinate descent; PG, projected gradient; MM, majorize–minimize.*

4 Variations on Newton's Method

The primary advantage of Newton's method is its speed of convergence in low-dimensional problems. Its many variants seek to retain its fast convergence while taming its defects. The variants all revolve around the core idea of locally approximating the objective function by a strictly convex quadratic. At each iteration, the quadratic approximation is optimized subject to safeguards that keep the iterates from overshooting and veering towards irrelevant stationary points.

Consider minimizing the real-valued function $f(\boldsymbol{\theta})$ defined on an open set $S \subset \mathrm{R}^p$. Assuming that $f(\boldsymbol{\theta})$ is twice differentiable, we have the second-order Taylor expansion

$$f(\boldsymbol{\gamma}) = f(\boldsymbol{\theta}) + df(\boldsymbol{\theta})(\boldsymbol{\gamma} - \boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\gamma} - \boldsymbol{\theta})^t d^2 f(\boldsymbol{\alpha})(\boldsymbol{\gamma} - \boldsymbol{\theta})$$

for some $\boldsymbol{\alpha}$ on the line segment $[\boldsymbol{\theta}, \boldsymbol{\gamma}]$. This expansion suggests that we substitute $d^2 f(\boldsymbol{\theta})$ for $d^2 f(\boldsymbol{\alpha})$ and approximate $f(\boldsymbol{\gamma})$ by the resulting quadratic. If we take this approximation seriously, then we can solve for its minimum point $\boldsymbol{\gamma}$ as

$$\boldsymbol{\gamma} = \boldsymbol{\theta} - d^2 f(\boldsymbol{\theta})^{-1} \nabla f(\boldsymbol{\theta}).$$

In Newton's method, we iterate according to

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - s d^2 f(\boldsymbol{\theta}_n)^{-1} \nabla f(\boldsymbol{\theta}_n) \tag{3}$$

for step length constant s with default value 1. Any stationary point of $f(\boldsymbol{\theta})$ is a fixed point of Newton's method.

There is nothing to prevent Newton's method from heading uphill rather than downhill. The first-order expansion

$$f(\boldsymbol{\theta}_{n+1}) = f(\boldsymbol{\theta}_n) - s\, df(\boldsymbol{\theta}_n) d^2 f(\boldsymbol{\theta}_n)^{-1} \nabla f(\boldsymbol{\theta}_n) + o(s)$$

makes it clear that the descent property holds provided $s > 0$ is small enough and the Hessian matrix $d^2 f(\boldsymbol{\theta}_n)$ is positive definite. When $d^2 f(\boldsymbol{\theta}_n)$ is not positive definite, it is usually replaced by a positive definite approximation $\boldsymbol{H}_n$ in the update (3).

Backtracking is crucial to avoid overshooting. In the step-halving version of backtracking, one starts with $s = 1$. If the descent property holds, then one takes the Newton step. Otherwise, $s/2$ is substituted for s, θ_{n+1} is recalculated, and the descent property is rechecked. Eventually, a small enough s is generated to guarantee $f(\boldsymbol{\theta}_{n+1}) < f(\boldsymbol{\theta}_n)$.

In the next two examples, we adopt standard statistical language. The outcome of a statistical experiment is summarized by a log likelihood $L(\boldsymbol{\theta})$. Its gradient $\nabla L(\boldsymbol{\theta})$ is called the score, and its second differential $d^2L(\boldsymbol{\theta})$, after a change in sign, is called the observed information. In maximum likelihood estimation, one maximizes $L(\boldsymbol{\theta})$ with respect to the parameter vector $\boldsymbol{\theta}$.

Example 4. *Newton's method for binomial regression*

Consider binomial regression with m independent responses $y_1, \ldots, y_m$. Each y_i represents a count between 0 and k_i with success probability $\pi_i(\boldsymbol{\theta})$ per trial. The log likelihood, score, and observed information amount to

$$\begin{aligned}
L(\boldsymbol{\theta}) &= \sum_{i=1}^{m} [y_i \ln \pi_i(\boldsymbol{\theta}) + (k_i - y_i) \ln[1 - \pi_i(\boldsymbol{\theta})] \\
\nabla L(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \frac{y_i - k_i \pi_i(\boldsymbol{\theta})}{\pi_i(\boldsymbol{\theta})[1 - \pi_i(\boldsymbol{\theta})]} \nabla \pi_i(\boldsymbol{\theta}) \\
-d^2 L(\boldsymbol{\theta}) &= -\sum_{i=1}^{m} \frac{y_i - k_i \pi_i(\boldsymbol{\theta})}{\pi_i(\boldsymbol{\theta})[1 - \pi_i(\boldsymbol{\theta})]} d^2 \pi_i(\boldsymbol{\theta}) \\
&\quad + \sum_{i=1}^{m} \left\{ \frac{y_i}{\pi_i(\boldsymbol{\theta})^2} + \frac{k_i - y_i}{[1 - \pi_i(\boldsymbol{\theta})]^2} \right\} \nabla \pi_i(\boldsymbol{\theta}) d\pi_i(\boldsymbol{\theta}).
\end{aligned}$$

Because $\mathrm{E}(y_i) = k_i \pi_i(\boldsymbol{\theta})$, the observed information can be approximated by

$$\begin{aligned}
-d^2 L(\boldsymbol{\theta}) &\approx \sum_{i=1}^{m} \left\{ \frac{y_i}{\pi_i(\boldsymbol{\theta})^2} + \frac{k_i - y_i}{[1 - \pi_i(\boldsymbol{\theta})]^2} \right\} \nabla \pi_i(\boldsymbol{\theta}) d\pi_i(\boldsymbol{\theta}) \\
&\approx \sum_{i=1}^{m} \left\{ \frac{k_i}{\pi_i(\boldsymbol{\theta})} + \frac{k_i}{[1 - \pi_i(\boldsymbol{\theta})]} \right\} \nabla \pi_i(\boldsymbol{\theta}) d\pi_i(\boldsymbol{\theta}).
\end{aligned}$$

Because we seek to maximize rather than minimize $L(\boldsymbol{\theta})$, we want $-d^2L(\boldsymbol{\theta})$ to be positive definite. Fortunately, both approximations fulfil this requirement. The second approximation leads to the scoring algorithm discussed later.

Example 5. *Poisson multigraph model*

In a graph, the number of edges between any two nodes is 0 or 1. A multigraph allows an arbitrary number of edges between any two nodes. Multigraphs are natural structures for modelling the internet and gene and protein networks. Here, we consider a multigraph with a random number of edges X_{ij} connecting every pair of nodes $\{i, j\}$. In particular, we assume that the X_{ij} are independent Poisson random variables with means μ_{ij}. As a plausible model for ranking nodes, we take $\mu_{ij} = \theta_i \theta_j$, where θ_i and θ_j are non-negative propensities (Ranola *et al.*, 2010). The log likelihood of the observed edge counts $x_{ij} = x_{ji}$ amounts to

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_{\{i,j\}} (x_{ij} \ln \mu_{ij} - \mu_{ij} - \ln x_{ij}!) \\ &= \sum_{\{i,j\}} [x_{ij}(\ln \theta_i + \ln \theta_j) - \theta_i \theta_j - \ln x_{ij}!]. \end{aligned}$$

The score vector has entries

$$\frac{\partial}{\partial \theta_i} L(\boldsymbol{\theta}) = \sum_{j \neq i} \left(\frac{x_{ij}}{\theta_i} - \theta_j \right),$$

and the observed information matrix has entries

$$-\frac{\partial^2}{\partial \theta_i \partial \theta_j} L(\boldsymbol{\theta}) = \begin{cases} 1 & i \neq j \\ \frac{1}{\theta_i^2} \sum_{k \neq i} x_{ik} & i = j. \end{cases}$$

For p nodes, the matrix $-d^2 L(\boldsymbol{p})$ is $p \times p$, and inverting it seems out of the question when p is large. Fortunately, the Sherman–Morrison formula comes to the rescue. If we write $-d^2 L(\boldsymbol{\theta})$ as $\boldsymbol{D} + \mathbf{1}\mathbf{1}^t$ with $\boldsymbol{D}$ diagonal, then the explicit inverse

$$(\boldsymbol{D} + \mathbf{1}\mathbf{1}^t)^{-1} = \boldsymbol{D}^{-1} - \frac{1}{1 + \mathbf{1}^t \boldsymbol{D}^{-1} \mathbf{1}} \boldsymbol{D}^{-1} \mathbf{1}\mathbf{1}^t \boldsymbol{D}^{-1}$$

is available. This makes Newton's method trivial to implement as long as one respects the bounds $\theta_i \geq 0$. More generally, it is always cheap to invert a low-rank perturbation of an explicitly invertible matrix.

In maximum likelihood estimation, the method of steepest ascent replaces the observed information matrix $-d^2 L(\boldsymbol{\theta})$ by the identity matrix $\boldsymbol{I}$. Fisher's scoring algorithm makes the far more effective choice of replacing the observed information matrix by the expected information matrix $J(\boldsymbol{\theta}) = E[-d^2 L(\boldsymbol{\theta})]$ (Osborne, 1992). The alternative representation $J(\boldsymbol{\theta}) = \text{Var}[\nabla L(\boldsymbol{\theta})]$ of $J(\boldsymbol{\theta})$ as a variance matrix demonstrates that it is positive semidefinite. Usually it is positive definite as well and serves as an excellent substitute for $-d^2 L(\boldsymbol{\theta})$ in Newton's method. The inverse matrices $-d^2 L(\hat{\boldsymbol{\theta}})^{-1}$ and $J(\hat{\boldsymbol{\theta}})^{-1}$ immediately supply the asymptotic variances and covariances of the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ (Rao, 1973).

The score and expected information simplify considerably for exponential families of densities (Bradley, 1973; Charnes *et al.*, 1976; Green, 1984; Jennrich & Moore, 1975; Nelder & Wedderburn, 1972). Recall that the density of a vector random variable $\boldsymbol{Y}$ from an exponential family can be written as

$$f(\boldsymbol{y} \mid \boldsymbol{\theta}) = g(\boldsymbol{y}) e^{\beta(\boldsymbol{\theta}) + h(\boldsymbol{y})^t \boldsymbol{\gamma}(\boldsymbol{\theta})} \tag{4}$$

relative to some measure ν (Dobson, 1990; Rao, 1973). The function $h(\boldsymbol{y})$ in (4) is the sufficient statistic. The maximum likelihood estimate of the parameter vector $\boldsymbol{\theta}$ depends on an observation $\boldsymbol{y}$ only through $h(\boldsymbol{y})$. Predictors of $\boldsymbol{y}$ are incorporated into the functions $\beta(\boldsymbol{\theta})$ and $\gamma(\boldsymbol{\theta})$. If $\gamma(\boldsymbol{\theta})$ is linear in $\boldsymbol{\theta}$, then $J(\boldsymbol{\theta}) = -d^2 L(\boldsymbol{\theta}) = -d^2 \beta(\boldsymbol{\theta})$, and scoring coincides with Newton's method. If in addition $J(\boldsymbol{\theta})$ is positive definite, then $L(\boldsymbol{\theta})$ is strictly concave and possesses at most a single local maximum, which is necessarily the global maximum.

Both the score vector and expected information matrix can be expressed succinctly in terms of the mean vector $\mu(\boldsymbol{\theta}) = \text{E}[h(\boldsymbol{y})]$ and the variance matrix $\Sigma(\boldsymbol{\theta}) = \text{Var}[h(\boldsymbol{y})]$ of the sufficient statistic. Standard arguments show that

$$\begin{aligned} \nabla L(\boldsymbol{\theta}) &= d\mu(\boldsymbol{\theta})^t \Sigma(\boldsymbol{\theta})^{-1} [h(\boldsymbol{y}) - \mu(\boldsymbol{\theta})] \\ J(\boldsymbol{\theta}) &= d\mu(\boldsymbol{\theta})^t \Sigma(\boldsymbol{\theta})^{-1} d\mu(\boldsymbol{\theta}). \end{aligned}$$

These formulas have had an enormous impact on non-linear regression and fitting generalized linear models. Applied statistics as we know it would be nearly impossible without them. Implementation of scoring is almost always safeguarded by step halving and upgraded to handle linear constraints and parameter bounds. The notion of quadratic approximation is still the key, but each step of constrained scoring must solve a quadratic programme.

In parallel with developments in statistics, numerical analysts sought substitutes for Newton's method. Their efforts a generation ago focused on quasi-Newton methods for generic smooth functions (Dennis & Schnabel, 1996; Nocedal & Wright, 2006). Once again, the core idea was successive quadratic approximation. A good quasi-Newton method (a) minimizes a quadratic function $f(\boldsymbol{\theta})$ from R^p to R in p steps, (b) avoids evaluation of $d^2 f(\boldsymbol{\theta})$, (c) adapts readily to simple parameter constraints, and (d) exploits inexact line searches.

Quasi-Newton methods update the current approximation $\boldsymbol{H}_n$ to the second differential $d^2 f(\boldsymbol{\theta})$ of an objective function $f(\boldsymbol{\theta})$ by a rank-one or rank-two perturbation satisfying a secant condition. The secant condition captures the first-order Taylor approximation

$$\nabla f(\boldsymbol{\theta}_{n+1}) - \nabla f(\boldsymbol{\theta}_n) \approx d^2 f(\boldsymbol{\theta}_n)(\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n).$$

If we define the gradient and argument differences

$$\begin{aligned} \boldsymbol{g}_n &= \nabla f(\boldsymbol{\theta}_{n+1}) - \nabla f(\boldsymbol{\theta}_n) \\ \boldsymbol{d}_n &= \boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n, \end{aligned}$$

then the secant condition reads $\boldsymbol{H}_{n+1}\boldsymbol{d}_n = \boldsymbol{g}_n$. Davidon (1959) discovered that the unique symmetric rank-one update to $\boldsymbol{H}_n$ satisfying the secant condition is

$$\boldsymbol{H}_{n+1} = \boldsymbol{H}_n + c_n \boldsymbol{v}_n \boldsymbol{v}_n^t,$$

where the constant c_n and the vector $\boldsymbol{v}_n$ are determined by

$$\begin{aligned} c_n &= -\frac{1}{(\boldsymbol{H}_n \boldsymbol{d}_n - \boldsymbol{g}_n)^t \boldsymbol{d}_n} \\ \boldsymbol{v}_n &= \boldsymbol{H}_n \boldsymbol{d}_n - \boldsymbol{g}_n. \end{aligned}$$

When the inner product $(\boldsymbol{H}_n \boldsymbol{d}_n - \boldsymbol{g}_n)^t \boldsymbol{d}_n$ is too close to 0, there are two possibilities. Either the secant adjustment is ignored, and the value $\boldsymbol{H}_n$ is retained for $\boldsymbol{H}_{n+1}$, or one resorts to a trust region strategy (Nocedal & Wright, 2006).

In the trust region method, one minimizes the quadratic approximation to $f(\boldsymbol{\theta})$ subject to the spherical constraint $\|\boldsymbol{\theta} - \boldsymbol{\theta}_n\|^2 \leq r^2$ for a fixed radius r. This constrained optimization problem has a solution regardless of whether $\boldsymbol{H}_n$ is positive definite. Working within a trust region prevents absurdly large steps in the early stages of minimization. With appropriate safeguards, some numerical analysts (Conn *et al.*, 1991; Khalfan *et al.*, 1993) consider Davidon's rank-one

update superior to the widely used BFGS update, named after Broyden, Fletcher, Goldfarb, and Shanno. This rank-two perturbation is guaranteed to maintain positive definiteness and is better understood theoretically than the symmetric rank-one update. Also of interest is the Davidon, Fletcher, and Powell (DFP) rank-two update, which applies to the inverse $\boldsymbol{H}_n^{-1}$ of $\boldsymbol{H}_n$. Although the DFP update ostensibly avoids matrix inversion, the consensus is that the BFGS update is superior to it in numerical practice (Dennis & Schnabel, 1996).

5 The MM and EM Algorithms

The numerical analysts Ortega and Rheinboldt (1970) first articulated the MM principle; de Leeuw (1977) saw its potential and created the first MM algorithm. The MM algorithm currently enjoys its greatest vogue in computational statistics (Hunter & Lange, 2004; Lange *et al.*, 2000; Wu & Lange, 2010). The basic idea is to convert a hard optimization problem into a sequence of simpler ones. In minimization, the MM principle majorizes the objective function $f(\boldsymbol{\theta})$ by a surrogate function $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ anchored at the current point $\boldsymbol{\theta}_n$. Majorization combines the tangency condition $g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n) = f(\boldsymbol{\theta}_n)$ and the domination condition $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n) \geq f(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$. The next iterate of the MM algorithm is defined to minimize $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$. Because

$$f(\boldsymbol{\theta}_{n+1}) \leq g(\boldsymbol{\theta}_{n+1} \mid \boldsymbol{\theta}_n) \quad \leq \quad g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n) \quad = \quad f(\boldsymbol{\theta}_n),$$

the MM iterates generate a descent algorithm driving the objective function downhill. Strictly speaking, the descent property depends only on decreasing $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$, not on minimizing it. Constraint satisfaction is automatically enforced in finding $\boldsymbol{\theta}_{n+1}$. Under appropriate regularity conditions, an MM algorithm is guaranteed to converge to a local minimum of the objective function (Lange, 2010). In maximization, we first minorize and then maximize. Thus, the acronym MM does double duty in the forms majorize–minimize and minorize–maximize.

When it is successful, the MM algorithm simplifies optimization by (a) separating the variables of a problem, (b) avoiding large matrix inversions, (c) linearizing a problem, (d) restoring symmetry, (e) dealing with equality and inequality constraints gracefully, and (f) turning a nondifferentiable problem into a smooth problem. The art in devising an MM algorithm lies in choosing a tractable surrogate function $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ that hugs the objective function $f(\boldsymbol{\theta})$ as tightly as possible.

The majorization relation between functions is closed under the formation of sums, nonnegative products, limits, and composition with an increasing function. These rules allow one to work piecemeal in simplifying complicated objective functions. Skill in dealing with inequalities is crucial in constructing majorizations. Classical inequalities such as Jensen's inequality, the information inequality, the arithmetic-geometric mean inequality, and the Cauchy–Schwartz prove useful in many problems. The supporting hyperplane property of a convex function and the quadratic upper bound principle of Böhning & Lindsay (1988) also find wide application.

Example 6. *An MM algorithm for non-negative least squares*

Sha *et al.* (2003) devised an MM algorithm for Example 1. The diagonal terms $a_{ii}\theta_i^2$ they retain as presented. The off-diagonal terms $a_{ij}\theta_i\theta_j$ they majorize according to the sign of the coefficient a_{ij}. When the sign of a_{ij} is positive, they apply the majorization

$$xy \leq \frac{y_n}{2x_n}x^2 + \frac{x_n}{2y_n}y^2,$$

which is just a rearrangement of the inequality

$$0 \le \left(\sqrt{\frac{y_n}{x_n}} x - \sqrt{\frac{x_n}{y_n}} y \right)^2,$$

with equality when $x = x_n$ and $y = y_n$. When the sign of a_{ij} is negative, they apply the majorization

$$-xy \le -x_n y_n \left[1 + \ln\left(\frac{x}{x_n}\right) + \ln\left(\frac{y}{y_n}\right) \right],$$

which is just a rearrangement of the simple inequality $z \ge 1 + \ln z$ with $z = xy/(x_n y_n)$. The value $z = 1$ gives equality in the inequality. Both majorizations separate parameters and allow one to minimize the surrogate function parameter by parameter. Indeed, if we define matrices $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ with entries $\max\{a_{ij}, 0\}$ and $-\min\{a_{ij}, 0\}$, respectively, then the resulting MM algorithm iterates according to

$$\theta_{n+1,i} = \theta_{n,i} \left[\frac{-b_i + \sqrt{b_i^2 + 4(\boldsymbol{A}^+ \boldsymbol{\theta}_n)_i (\boldsymbol{A}^- \boldsymbol{\theta}_n)_i}}{2(\boldsymbol{A}^+ \boldsymbol{\theta}_n)_i} \right].$$

All entries of the initial point $\boldsymbol{\theta}_0$ should be positive; otherwise, the MM algorithm stalls. The updates occur in parallel. In contrast, the cyclic coordinate descent updates are sequential. Figure 1 depicts the progress of the MM algorithm on our non-negative least squares problem.

Example 7. *Locating a gunshot*

Locating the time and place of a gunshot is a typical global positioning problem (Strang & Borre, 2012). In a certain city, m sensors located at the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ are installed. A signal, say a gunshot sound, is sent from an unknown location $\boldsymbol{\theta}$ at unknown time α and known speed s and arrives at location j at time y_j observed with random measurement error. The problem is to estimate the vector $\boldsymbol{\theta}$ and the scalar α from the observed data $y_1, \ldots, y_m$. Other problems of this nature include pinpointing the epicentre of an earthquake and the detonation point of a nuclear explosion. This estimation problem can be attacked by a combination of block descent and the MM principle.

If we assume Gaussian random errors, then maximum likelihood estimation reduces to minimizing the criterion

$$\begin{aligned} f(\boldsymbol{\theta}, \alpha) &= \frac{1}{2} \sum_{j=1}^{m} \left(y_j - s^{-1} \|\boldsymbol{\theta} - \boldsymbol{x}_j\| - \alpha \right)^2 \\ &= \frac{1}{2s^2} \sum_{j=1}^{m} \left(s y_j - \|\boldsymbol{\theta} - \boldsymbol{x}_j\| - \alpha s \right)^2. \end{aligned}$$

The equivalence of the two representations of $f(\boldsymbol{\theta}, \alpha)$ shows that it suffices to solve the problem with speed $s = 1$. In the remaining discussion, we make this assumption. For a fixed $\boldsymbol{\theta}$, estimation of α reduces to a least squares problem with the obvious solution

$$\alpha = \frac{1}{m} \sum_{j=1}^{m} \left(y_j - \|\boldsymbol{\theta} - \boldsymbol{x}_j\| \right).$$

To update $\boldsymbol{\theta}$ with a fixed α, we rewrite $f(\boldsymbol{\theta}, \alpha)$ as

$$f(\boldsymbol{\theta}, \alpha) = \frac{1}{2} \sum_{j=1}^{m} \left[(y_j - \alpha)^2 - 2(y_j - \alpha) \|\boldsymbol{\theta} - \boldsymbol{x}_j\| + \|\boldsymbol{\theta} - \boldsymbol{x}_j\|^2 \right].$$

The middle terms $-2(y_j - \alpha)\|\boldsymbol{\theta} - \boldsymbol{x}_j\|$ are awkward to deal with in minimization. Depending on the sign of the coefficient $-2(y_j - \alpha)$, we majorized them in two different ways. If the sign is negative, then we employ the Cauchy–Schwarz majorization

$$-\|\boldsymbol{\theta} - \boldsymbol{x}_j\| \leq -\|\boldsymbol{\theta}_n - \boldsymbol{x}_j\|^{-1} (\boldsymbol{\theta}_n - \boldsymbol{x}_j)^t (\boldsymbol{\theta} - \boldsymbol{x}_j).$$

If the sign is positive, then we employ the more subtle majorization

$$\|\boldsymbol{\theta} - \boldsymbol{x}_j\| \leq \|\boldsymbol{\theta}_n - \boldsymbol{x}_j\| + \frac{1}{2\|\boldsymbol{\theta}_n - \boldsymbol{x}_j\|} \left(\|\boldsymbol{\theta} - \boldsymbol{x}_j\|^2 - \|\boldsymbol{\theta}_n - \boldsymbol{x}_j\| \right).$$

To derive this second majorization, note that $\sqrt{u}$ is a concave function on $(0, \infty)$. It therefore satisfies the dominating hyperplane inequality

$$\sqrt{u} \leq \sqrt{u_n} + \frac{1}{2\sqrt{u_n}}(u - u_n).$$

Now substitute $\|\boldsymbol{\theta} - \boldsymbol{x}_j\|^2$ for u. These manoeuvres separate parameters and reduce the surrogate to a sum of linear terms and squared Euclidean norms. The minimization of the surrogate yields the MM update

$$\boldsymbol{\theta}_{n+1} = \frac{\sum_{j=1}^{m} \left[1 + \frac{(\alpha - y_j)}{\|\boldsymbol{\theta}_n - \boldsymbol{x}_j\|} \right] \boldsymbol{x}_j - \left[\sum_{\alpha \leq y_j} \frac{(\alpha - y_j)}{\|\boldsymbol{\theta}_n - \boldsymbol{x}_j\|} \right] \boldsymbol{\theta}_n}{m + \sum_{\alpha > y_j} \frac{\alpha - y_j}{\|\boldsymbol{\theta}_n - \boldsymbol{x}_j\|}}$$

of $\boldsymbol{\theta}$ for a fixed α. The condition $\alpha > y_j$ in this update is usually vacuous. By design, $f(\boldsymbol{\theta}, \alpha)$ decreases after each cycle of updating α and $\boldsymbol{\theta}$.

The celebrated EM algorithm is one the most potent optimization tools in the statistician's toolkit (Dempster *et al.*, 1977; McLachlan & Krishnan, 2008). The E step in the EM algorithm creates a surrogate function, the Q function in the literature, that minorizes the log likelihood. Thus, every EM algorithm is an MM algorithm. If $\boldsymbol{y}$ is the observed data and $\boldsymbol{x}$ is the complete data, then the Q function is defined as the conditional expectation

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n) = \mathrm{E}[\ln f(\boldsymbol{X} \mid \boldsymbol{\theta}) \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}_n],$$

where $f(\boldsymbol{x} \mid \boldsymbol{\theta})$ denotes the complete data log likelihood, upper case letters indicate random vectors, and lower case letters indicate corresponding realizations of these random vectors. In the M step of the EM algorithm, one calculates the next iterate $\boldsymbol{\theta}_{n+1}$ by maximizing $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ with respect to $\boldsymbol{\theta}$.

Example 8. *MM versus EM for the Dirichlet-multinomial distribution*

When multivariate count data exhibit overdispersion, the Dirichlet-multinomial distribution is preferred to the multinomial distribution. In the Dirichlet-multinomial model, the multinomial probabilities $\boldsymbol{p} = (p_1, \ldots, p_d)$ follow a Dirichlet distribution with parameter vector $\boldsymbol{\alpha} =$

$(\alpha_1, \ldots, \alpha_d)$ having positive components. For a multivariate count vector $\boldsymbol{x} = (x_1, \ldots, x_d)$ with batch size $|\boldsymbol{x}| = \sum_{j=1}^d x_j$, the probability mass function is accordingly

$$\begin{aligned} h(\boldsymbol{x} \mid \boldsymbol{\alpha}) &= \int_{\Delta_d} \binom{|\boldsymbol{x}|}{\mathbf{x}} \prod_{j=1}^d p_j^{x_j} \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{j=1}^d \Gamma(\alpha_j)} \prod_{j=1}^d p_j^{\alpha_j - 1} \, d\boldsymbol{p} \\ &= \binom{|\boldsymbol{x}|}{\boldsymbol{x}} \frac{\prod_{j=1}^d \Gamma(\alpha_j + x_j)}{\Gamma(|\boldsymbol{\alpha}| + |\boldsymbol{x}|)} \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{j=1}^d \Gamma(\alpha_j)} \\ &= \binom{|\boldsymbol{x}|}{\boldsymbol{x}} \frac{\prod_{j=1}^d (\alpha_j)_{x_j}}{(|\boldsymbol{\alpha}|)_{|\boldsymbol{x}|}}, \end{aligned} \tag{5}$$

where Δ_d is the unit simplex in d dimensions, $|\boldsymbol{\alpha}|$ equals $\sum_{j=1}^d \alpha_j$, and $(a)_k = \prod_{i=0}^{k-1}(a+i)$ denotes a rising factorial. The last equality in (6) follows from the factorial property $\Gamma(a+1)/\Gamma(a) = a$ of the gamma function. Given independent data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, the log likelihood is

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^m \ln \binom{|\boldsymbol{x}_i|}{\boldsymbol{x}_i} + \sum_{i=1}^m \sum_{j=1}^d \sum_{k=0}^{x_{ij}-1} \ln(\alpha_j + k) - \sum_{i=1}^m \sum_{k=0}^{|\boldsymbol{x}_i|-1} \ln(|\boldsymbol{\alpha}| + k).$$

The lack of concavity of $L(\boldsymbol{\alpha})$ may cause instability in Newton's method when it is started far from the optimal point. Fisher's scoring algorithm is computationally prohibitive because calculation of the expected information matrix involves numerous evaluations of beta-binomial tail probabilities. The ascent property makes EM and MM algorithms attractive.

In deriving an EM algorithm, we treat the unobserved multinomial probabilities p_j in each case as missing data. The complete data likelihood is then the integrand in the integral (5). A straightforward calculation shows that $\boldsymbol{p}$ possesses a posterior Dirichlet distribution with parameters $\alpha_1 + x_{i1}$ through $\alpha_d + x_{id}$ for case i. If we now differentiate the identity

$$1 = \frac{\Gamma(|\boldsymbol{\alpha}| + |\boldsymbol{x}_i|)}{\prod_{j=1}^d \Gamma(\alpha_j + x_{ij})} \int_{\Delta_d} \prod_{j=1}^d p_j^{x_{ij} + \alpha_j - 1} d\boldsymbol{p}$$

with respect to α_j, then the identity

$$\mathrm{E}(\ln p_j \mid \boldsymbol{\alpha}_n, x_{ij}) = \Psi(x_{ij} + \alpha_{nj}) - \Psi(|\boldsymbol{x}_i| + |\boldsymbol{\alpha}_n|)$$

emerges, where $\Psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function. It follows that up to an irrelevant additive constant, the surrogate function is

$$\begin{aligned} Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}_n) = &\sum_{i=1}^m \sum_{j=1}^d \alpha_j \left[\Psi(x_{ij} + \alpha_{nj}) - \Psi(|\boldsymbol{x}_i| + |\boldsymbol{\alpha}_n|)\right] \\ &+ m \ln \Gamma(|\boldsymbol{\alpha}|) - m \sum_{j=1}^d \ln \Gamma(\alpha_j). \end{aligned}$$

Maximizing $Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}_n)$ is non-trivial because it involves special functions and intertwining of the α_j parameters.

Directly invoking the MM principle produces a more malleable surrogate function (Zhou & Lange, 2010). Consider the logarithm of the third form of the likelihood function (5). Applying Jensen's inequality to $\ln(\alpha_j + k)$ gives

$$\begin{aligned}\ln(\alpha_j + k) &\geq \frac{\alpha_{nj}}{\alpha_{nj} + k} \ln\left(\frac{\alpha_{nj} + k}{\alpha_{nj}} \cdot \alpha_j\right) + \frac{k}{\alpha_{nj} + k} \ln\left(\frac{\alpha_{nj} + k}{k} \cdot k\right) \\ &= \frac{\alpha_{nj}}{\alpha_{nj} + k} \ln \alpha_j + c_n.\end{aligned}$$

Likewise, applying the supporting hyperplane inequality to $-\ln(|\boldsymbol{\alpha}| + k)$ gives

$$-\ln(|\boldsymbol{\alpha}| + k) \geq -\ln(|\boldsymbol{\alpha}_n| + k) - \frac{|\boldsymbol{\alpha}| - |\boldsymbol{\alpha}_n|}{|\boldsymbol{\alpha}_n| + k} = -\frac{|\boldsymbol{\alpha}|}{|\boldsymbol{\alpha}_n| + k} + c_n.$$

Overall, these minorizations yield the surrogate function

$$\begin{aligned}g(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}_n) &= -\sum_k \frac{r_k}{|\boldsymbol{\alpha}_n| + k} |\boldsymbol{\alpha}| + \sum_j \sum_k \frac{s_{jk}\alpha_{nj}}{\alpha_{nj} + k} \ln \alpha_j + c_n, \\ s_{jk} &= \sum_{i=1}^m 1_{\{x_{ij} \geq k+1\}}, \quad r_k = \sum_{i=1}^m 1_{\{m_i \geq k+1\}},\end{aligned}$$

which completely separates the parameter α_j. This suggests the simple MM updates

$$\alpha_{n+1,j} = \alpha_{nj} \frac{\sum_k \frac{s_{jk}}{\alpha_{nj}+k}}{\sum_k \frac{r_k}{|\boldsymbol{\alpha}_n|+k}}, \quad j = 1, \ldots, d.$$

The positivity constraints are always satisfied when all initial values $\alpha_{0j} > 0$. Parameter separation can be achieved in the EM algorithm by a further minorization of the $\ln \Gamma(|\boldsymbol{\alpha}|)$ term in $Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}_n)$. This action yields a viable EM–MM hybrid algorithm. The study of Zhou & Yang (2012) contains more details and a comparison of the convergence rates of the three algorithms.

Finally, let us mention various strategies for handling exceptional cases. In the MM algorithm, it may be impossible to optimize the surrogate function $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ explicitly. There are two obvious remedies. One is to institute some form of block relaxation in updating $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ (Meng & Rubin, 1993). There is no need to iterate to convergence because the purpose is merely to improve $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ and hence the objective function $f(\boldsymbol{\theta})$. Another obvious remedy is to optimize the surrogate function by Newton's method. It turns out that a single step of Newton's method suffices to preserve the local rate of convergence of the MM algorithm (Lange, 1995a). The ascent property is sacrificed initially, but it kicks in as one approaches the optimal point. In an unconstrained problem, this variant MM algorithm can be phrased as

$$\begin{aligned}\boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n + d^2 g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n)^{-1} \nabla g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n) \\ &= \boldsymbol{\theta}_n + d^2 g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n)^{-1} \nabla f(\boldsymbol{\theta}_n),\end{aligned}$$

where the substitution of $\nabla f(\boldsymbol{\theta}_n)$ for $\nabla g(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}_n)$ is justified by the tangency and domination conditions satisfied by $g(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n)$ and $f(\boldsymbol{\theta})$.

A more pressing concern in the EM algorithm is intractability of the E step. If $f(\boldsymbol{X} \mid \boldsymbol{\theta})$ denotes the complete data likelihood, then in the stochastic EM algorithm (Jank, 2006; Robert

& Casella, 2004; Wei & Tanner, 1990), one estimates the surrogate function by a Monte Carlo average

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_n) \approx \frac{1}{m} \sum_{i=1}^{m} \ln f(\boldsymbol{x}_i \mid \boldsymbol{\theta}) \tag{6}$$

over realizations $\boldsymbol{x}_i$ of the complete data $\boldsymbol{X}$ conditional on the observed data $\boldsymbol{Y} = \boldsymbol{y}$ and the current parameter iterate $\boldsymbol{\theta}_n$. Sampling can be performed by rejection sampling, importance sampling, Markov chain Monte Carlo, or quasi-Monte Carlo. The next iterate $\boldsymbol{\theta}_{n+1}$ should maximize the average (6). The sample size m should increase as the iteration count n increases. Determining the rate of increase of m and setting a reasonable convergence criterion are both subtle issues. The ascent property of the EM algorithm fails because of the inherent sampling noise. The combination of slow convergence and Monte Carlo sampling makes the stochastic EM algorithm unattractive in large-scale problems. In smaller problems, it fills a useful niche.

The stochastic EM algorithm generalizes the Robbins–Monro algorithm (Robbins & Monro, 1951) for root finding and the Kiefer–Wolfowitz algorithm (Kiefer & Wolfowitz, 1952) for function maximization. In unconstrained maximum likelihood estimation, one seeks a root of the likelihood equation, so both methods are relevant. Under suitable assumptions, the Kiefer–Wolfowitz algorithm converges to a local maximum almost surely. Because this cluster of topics is tangential to our overall emphasis on deterministic methods of optimization, we refer readers to the books of Chen (2002), Kushner & Yin (2003), and Robert & Casella (2004) for a fuller discussion.

6 Penalization

Penalization is a device for imposing parsimony. For purposes of illustration, we discuss two penalized estimation problems of considerable utility in applied statistics. Both of these examples generate convex programmes with non-differentiable objective functions. In the interests of accessibility, we will derive estimation algorithms for both problems without invoking the machinery of convex analysis.

Example 9. *Lasso penalized regression*

Lasso penalized regression has been pursued for a long time in many application areas (Chen *et al.*, 1998; Claerbout & Muir, 1973; Donoho & Johnstone, 1994; Santosa & Symes, 1986; Taylor *et al.*, 1979; Tibshirani, 1996). Modern versions consider a generalized linear model where y_i is the response for case i, x_{ij} is the value of predictor j for case i, and θ_j is the regression coefficient corresponding to predictor j. When the number of predictors p exceeds the number of cases m, $\boldsymbol{\theta}$ cannot be uniquely estimated. In an era of big data, this quandary is fairly common. One remedy is to perform model selection by imposing a lasso penalty on the loss function $\ell(\boldsymbol{\theta})$. In least squares estimation,

$$\ell(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{m} \left(y_i - \sum_j x_{ij} \theta_j \right)^2 .$$

For a generalized linear model (Park & Hastie, 2007), $\ell(\boldsymbol{\theta})$ is the negative log likelihood of the data. Lasso penalized estimation minimizes the criterion

$$f(\boldsymbol{\theta}) = \ell(\theta) + \rho \sum_j w_j |\theta_j|,$$

where the non-negative weights w_j and the tuning constant $\rho > 0$ are given. If θ_j is the intercept for the model, then its weight w_j is usually set to 0. For the remaining predictors, the choice $w_j = 1$ is reasonable provided the predictors are standardized to have mean 0 and variance 1. To improve the asymptotic properties of the lasso estimates, the adaptive lasso (Zou, 2006) defines the weights $w_j = |\hat{\theta}_j|^{-1}$ for any consistent estimate $\hat{\theta}_j$ of θ_j. In a Bayesian context, imposing a lasso penalty is equivalent to placing a Laplace prior with mean 0 on each θ_j. The elastic net adds a ridge penalty $\lambda \sum_j \theta_j^2$ to the lasso penalty (Zou & Hastie, 2005).

The primary difference between lasso and ridge regression is that the lasso penalty forces most parameters to 0, whereas the ridge penalty merely reduces them. Thus, the ridge penalty relaxes its grip too quickly for model selection. Unfortunately, the lasso penalty tends to select one predictor from a group of correlated predictors and ignore the others. The elastic net ameliorates this defect. To overcome severe shrinkage, many statisticians discard penalties after the conclusion of model selection and re-estimate the selected parameters. Cross-validation and stability selection are effective in choosing the penalty tuning constant and the selected predictors, respectively (Hastie *et al.*, 2009; Meinshausen & Bühlmann, 2010).

Coordinate descent works particularly well when only a few predictors enter a model (Friedman *et al.*, 2007; Wu & Lange, 2008). Consider what happens when we visit parameter θ_j and the loss function is the least squares criterion. If we define the amended response $z_{ni} = y_i - \sum_{k \neq j} x_{ik}\theta_{nk}$, then the problem reduces to minimizing

$$\frac{1}{2}\sum_{i=1}^{m}(z_{ni} - x_{ij}\theta_j)^2 + \rho w_j|\theta_j|.$$

Now divide the domain of θ_j into the two intervals $(-\infty, 0]$ and $[0, \infty)$. On the right interval, elementary calculus suggests the update

$$\theta_{n+1,j} = \frac{\sum_{i=1}^{m} z_{ni}x_{ij} - \rho w_j}{\sum_{i=1}^{m} x_{ij}^2}.$$

This is invalid when it is negative and must be replaced by 0. Likewise, on the left interval, we have the update

$$\theta_{n+1,j} = \frac{\sum_{i=1}^{m} z_{ni}x_{ij} + \rho w_j}{\sum_{i=1}^{m} x_{ij}^2}$$

unless it is positive. On both intervals, shrinkage pulls the usual least squares estimate towards 0. In underdetermined problems with just a few relevant predictors, most parameters never budge from their starting values of 0. This circumstance plus the complete absence of matrix operations explains the speed of coordinate descent. It inherits its numerical stability from the descent property enjoyed by any coordinate descent algorithm.

With a generalized linear model, say logistic regression, the same story plays out. Now, however, we must institute a line search for the minimum on each of the two half-intervals. Newton's method, scoring, and even golden section search work well. When $f(\boldsymbol{\theta})$ is convex, and $\theta_j = 0$, it is prudent to check the forward directional derivatives $d_{\boldsymbol{e}_j} f(\boldsymbol{\theta})$ and $d_{-\boldsymbol{e}_j} f(\boldsymbol{\theta})$ along the current coordinate direction $\boldsymbol{e}_j$ and its negative. If both forward directional derivatives are non-negative, then no progress can be made by moving off 0. Thus, a parameter parked at 0 is left there. Other computational savings are possible that make coordinate descent even faster. For example, computations can be organized around the linear predictor $\sum_j x_{ij}\theta_j$ for each case i. When θ_j changes, it is trivial to update this inner product. Wu *et al.* (2009) and Wu & Lange (2008) illustrate the potential of coordinate descent on some concrete genetic examples.

Example 10. *Matrix completion*

The matrix completion problem became famous when the movie distribution company Netflix offered a million dollar prize for improvements to its movie rating system (ACM SIGKDD and Netflix, 2007). The idea was that customers would submit ratings on a small subset of movie titles, and from these ratings, Netflix would infer their preferences and recommend additional movies for their consideration. Imagine therefore a very sparse matrix $\boldsymbol{Y} = (y_{ij})$ whose rows are individuals and whose columns are movies. Completed cells contain a rating from 1 to 5. Most cells are empty and need to be filled in. If the matrix is sufficiently structured and possesses low rank, then it is possible to complete the matrix in a parsimonious way. Although this problem sounds specialized, it has applications far beyond this narrow setting. For example, filling in missing genotypes in genome scans for disease genes benefits from matrix completion (Chi *et al.*, 2013).

Following Cai *et al.* (2008), Candés & Tao (2009), Mazumder *et al.* (2010), and Chen *et al.* (2012), let Δ denote the set of index pairs (i, j) such that y_{ij} is observed. The Lagrangian formulation of matrix completion minimizes the criterion

$$f(\boldsymbol{X}) = \frac{1}{2} \sum_{(i,j)\in\Delta} (y_{ij} - x_{ij})^2 + \rho \sum_k \sigma_k \tag{7}$$

with respect to a compatible matrix $\boldsymbol{X} = (x_{ij})$ with singular values σ_k. Recall that the singular value decomposition

$$\boldsymbol{X} = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^t$$

represents $\boldsymbol{X}$ as a sum of outer products involving a collection of orthogonal left singular vectors $\boldsymbol{u}_i$, a corresponding collection of orthogonal right singular vectors $\boldsymbol{v}_i$, and a descending sequence of non-negative singular values σ_i. Alternatively, we can factor $\boldsymbol{X}$ in the form $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^t$ for orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ and a rectangular diagonal matrix $\boldsymbol{\Sigma}$.

The nuclear norm $\|\boldsymbol{X}\|_{\text{nuc}} = \sum_k \sigma_k$ plays the same role in low-rank matrix approximation that the ℓ_1 norm $\|\boldsymbol{b}\|_1 = \sum_k |b_k|$ plays in sparse regression. For a more succinct representation of the criterion (7), we introduce the Frobenius norm

$$\|\boldsymbol{U}\|_{\text{F}} = \sqrt{\sum_i \sum_j u_{ij}^2}$$

induced by the trace inner product $\text{tr}(\boldsymbol{U}\boldsymbol{V}^t)$ and the projection operator $P_\Delta(\boldsymbol{Y})$ with entries

$$P_\Delta(\boldsymbol{Y}) = \begin{cases} y_{ij} & (i,j) \in \Delta \\ 0 & (i,j) \notin \Delta. \end{cases}$$

In this notation, the criterion (7) becomes

$$\frac{1}{2}\|P_\Delta(\boldsymbol{Y}) - P_\Delta(\boldsymbol{X})\|_F^2 + \rho\|\boldsymbol{X}\|_{\text{nuc}}.$$

To derive an algorithm for estimating $\boldsymbol{X}$, we again exploit the MM principle. The general idea is to restore the symmetry of the problem by imputing the missing data (Mazumder *et al.*, 2010). Suppose $\boldsymbol{X}_n$ is our current approximation to $\boldsymbol{X}$. We simply replace a missing entry y_{ij} of $\boldsymbol{Y}$ by the corresponding entry x_{nij} of $\boldsymbol{X}_n$ and add the term $1/2\ (x_{nij} - x_{ij})^2$ to the

criterion (7). Because the added terms majorize 0, they create a legitimate surrogate function and lead to an MM algorithm. One can rephrase the problem in matrix terms by defining the orthogonal complement $P_\Delta^\perp(\boldsymbol{Y})$ of $P_\Delta(\boldsymbol{Y})$ according to the rule $P_\Delta^\perp(\boldsymbol{Y}) + P_\Delta(\boldsymbol{Y}) = \boldsymbol{Y}$. The matrix $\boldsymbol{Z}_n = P_\Delta(\boldsymbol{Y}) + P_\Delta^\perp(\boldsymbol{X}_n)$ temporarily completes $\boldsymbol{Y}$ and yields the surrogate function

$$\begin{aligned} g(\boldsymbol{X} \mid \boldsymbol{X}_n) &= \frac{1}{2}\|\boldsymbol{Z}_n - \boldsymbol{X}\|_{\mathrm{F}}^2 + \rho\|\boldsymbol{X}\|_{\mathrm{nuc}} \\ &= \frac{1}{2}\|\boldsymbol{Z}_n\|_{\mathrm{F}}^2 - \mathrm{tr}(\boldsymbol{Z}_n\boldsymbol{X}^t) + \frac{1}{2}\|\boldsymbol{X}\|_{\mathrm{F}}^2 + \rho\|\boldsymbol{X}\|_{\mathrm{nuc}}. \end{aligned}$$

At this juncture, it is helpful to recall some mathematical facts. First, the Frobenius norm is invariant under left and right multiplication of its argument by an orthogonal matrix. Thus, $\|\boldsymbol{X}\|_{\mathrm{F}}^2 = \sum_k \sigma_k^2$ depends only on the singular values of $\boldsymbol{X}$. The inner product $-\mathrm{tr}(\boldsymbol{Z}_n\boldsymbol{X}^t)$ presents a greater barrier to progress, but it ultimately succumbs to a matrix analogue of the Cauchy–Schwarz inequality. Fan's inequality says that

$$\mathrm{tr}(\boldsymbol{Z}_n\boldsymbol{X}^t) \le \sum_k \omega_k\sigma_k$$

for the ordered singular values ω_k of $\boldsymbol{Z}_n$ (Borwein & Lewis, 2000). Equality is attained in Fan's inequality if and only if the right and left singular vectors for the two matrices coincide. Thus, in minimizing $g(\boldsymbol{X} \mid \boldsymbol{X}_n)$, we can assume that the singular vectors of $\boldsymbol{X}$ coincide with those of $\boldsymbol{Z}_n$ and rewrite the surrogate function as

$$\begin{aligned} g(\boldsymbol{X} \mid \boldsymbol{X}_n) &= \frac{1}{2}\sum_k \omega_k^2 - \sum_k \omega_k\sigma_k + \frac{1}{2}\sum_k \sigma_k^2 + \rho\sum_k \sigma_k \\ &= \frac{1}{2}\sum_k (\omega_k - \sigma_k)^2 + \rho\sum_k \sigma_k. \end{aligned}$$

Application of the forward directional derivative test

$$d_{\boldsymbol{v}}\left[\frac{1}{2}\sum_k (\omega_k - \sigma_k)^2 + \rho\sum_k \sigma_k\right] = \sum_k (\sigma_k - \omega_k)v_k + \rho\sum_k v_k \ge 0$$

for all tangent directions $\boldsymbol{v}$ identifies the shrunken singular values

$$\sigma_k = \max\{\omega_k - \rho, 0\}$$

as optimal. In practice, one does not have to extract the full singular value decomposition of $\boldsymbol{Z}_n$. Only the singular values $\omega_k > \rho$ are actually relevant in constructing $\boldsymbol{X}_{n+1}$.

In many applications, the underlying structure of the observation matrix $\boldsymbol{Y}$ is corrupted by a few noisy entries. This tempts one to approximate $\boldsymbol{Y}$ by the sum of a low-rank matrix $\boldsymbol{X}$ plus a sparse matrix $\boldsymbol{W}$. To estimate $\boldsymbol{X}$ and $\boldsymbol{W}$, we introduce a positive tuning constant λ and minimize the criterion

$$f(\boldsymbol{X}, \boldsymbol{W}) = \frac{1}{2}\sum_{(i,j)\in\Delta} (y_{ij} - x_{ij} - w_{ij})^2 + \rho\sum_k \sigma_k + \lambda\sum_i\sum_j |w_{ij}|$$

by block descent. We have already indicated how to update $\boldsymbol{X}$ for a fixed $\boldsymbol{W}$. To minimize $f(\boldsymbol{X}, \boldsymbol{W})$ for a fixed $\boldsymbol{X}$, we set $w_{ij} = 0$ for any pair $(i, j) \notin \Delta$. Because the remaining $\boldsymbol{W}$ parameters separate in $f(\boldsymbol{X}, \boldsymbol{W})$, the shrinkage updates

$$w_{n+1,i,j} = \begin{cases} y_{ij} - x_{nij} - \lambda & y_{ij} - x_{nij} - \lambda > 0 \\ y_{ij} - x_{nij} + \lambda & y_{ij} - x_{nij} + \lambda < 0 \\ 0 & \text{otherwise} \end{cases}$$

are trivial to derive.

7 Augmented Lagrangians

The augmented Lagrangian method is one of the best ways of handling parameter constraints (Hestenes, 1969; Nocedal & Wright, 2006; Powell, 1969; Rockafellar, 1973). For the sake of simplicity, we focus on the problem of minimizing $f(\boldsymbol{\theta})$ subject to the equality constraints $g_i(\boldsymbol{\theta}) = 0$ for $i = 1, \ldots, q$. We will ignore inequality constraints and assume that $f(\boldsymbol{\theta})$ and the $g_i(\boldsymbol{\theta})$ are smooth. At a constrained minimum, the classical Lagrange multiplier rule

$$\mathbf{0} = \nabla f(\boldsymbol{\theta}) + \sum_{i=1}^{q} \lambda_i \nabla g_i(\boldsymbol{\theta}) \tag{8}$$

holds provided the gradients $\nabla g_i(\boldsymbol{\theta})$ are linearly independent. The augmented Lagrangian method optimizes the perturbed function

$$\mathcal{L}_\rho(\boldsymbol{\theta}, \boldsymbol{\lambda}) = f(\boldsymbol{\theta}) + \sum_{i=1}^{q} \lambda_i g_i(\boldsymbol{\theta}) + \frac{\rho}{2} \sum_{i=1}^{q} g_i(\boldsymbol{\theta})^2$$

with respect to $\boldsymbol{\theta}$. It then adjusts the current multiplier vector $\boldsymbol{\lambda}$ in the hope of matching the true Lagrange multiplier vector. The penalty term $\rho/2\ g_i(\boldsymbol{\theta})^2$ punishes violations of the equality constraint $g_i(\boldsymbol{\theta}) = 0$. At convergence, the gradient $\rho g_i(\boldsymbol{\theta}) \nabla g_i(\boldsymbol{\theta})$ of $\rho/2\ g_i(\boldsymbol{\theta})^2$ vanishes, and we recover the standard multiplier rule (8). This process can only succeed if the degree of penalization ρ is sufficiently large.

Thus, we must either take ρ initially large or gradually increase it until it hits the finite transition point where the constrained and unconstrained solutions merge. Updating $\boldsymbol{\lambda}$ is more subtle. If $\boldsymbol{\theta}_n$ furnishes the unconstrained minimum of $\mathcal{L}_\rho(\boldsymbol{\theta}, \boldsymbol{\lambda}_n)$, then the stationarity condition reads

$$\begin{aligned} 0 &= \nabla f(\boldsymbol{\theta}_n) + \sum_{i=1}^{q} \lambda_{ni} \nabla g_i(\boldsymbol{\theta}_n) + \rho \sum_{i=1}^{q} g_i(\boldsymbol{\theta}_n) \nabla g_i(\boldsymbol{\theta}_n), \\ &= \nabla f(\boldsymbol{\theta}_n) + \sum_{i=1}^{q} [\lambda_{ni} + \rho g_i(\boldsymbol{\theta}_n)] \nabla g_i(\boldsymbol{\theta}_n). \end{aligned}$$

The last equation motivates the standard update

$$\lambda_{n+1,i} = \lambda_{ni} + \rho g_i(\boldsymbol{\theta}_n).$$

The alternating direction method of multipliers (ADMM) (Gabay & Mercier, 1976; Glowinski & Marrocco, 1975) minimizes the sum $f(\boldsymbol{\theta}) + h(\boldsymbol{\gamma})$ subject to the affine constraints

$\boldsymbol{A\theta} + \boldsymbol{B\gamma} = \boldsymbol{c}$. Although the objective function is separable in the block variables $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, the affine constraints frustrate a direct attack. However, the problem is ripe for a combination of the augmented Lagrangian method and a single round of block descent per iteration. The augmented Lagrangian is

$$\mathcal{L}_\rho(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) = f(\boldsymbol{\theta}) + h(\boldsymbol{\gamma}) + \boldsymbol{\lambda}^t(\boldsymbol{A\theta} + \boldsymbol{B\gamma} - \boldsymbol{c}) + \frac{\rho}{2}\|\boldsymbol{A\theta} + \boldsymbol{B\gamma} - \boldsymbol{c}\|^2.$$

Minimization is performed over $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ by block descent before updating the multiplier vector $\boldsymbol{\lambda}$ via

$$\boldsymbol{\lambda}_{n+1} = \boldsymbol{\lambda}_n + \rho(\boldsymbol{A\theta}_{n+1} + \boldsymbol{B\gamma}_{n+1} - \boldsymbol{c}).$$

Introduction of block descent simplifies the usual augmented Lagrangian method, which minimizes $\mathcal{L}_\rho(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\lambda})$ jointly over $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$. This modest change keeps the convergence theory intact (Boyd *et al.*, 2011; Fortin & Glowinski, 1983) and has led to a resurgence in the popularity of ADMM in machine learning (Bien & Tibshirani, 2011; Boyd *et al.*, 2011; Chen *et al.*, 2012; Qin & Goldfarb, 2012; Richard *et al.*, 2012; Xue *et al.*, 2012).

Example 11. *Fused lasso*

The ADMM is helpful in reducing difficult optimization problems to simpler ones. The easiest fused lasso problem minimizes the criterion (Tibshirani *et al.*, 2005)

$$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\theta}\|_2^2 + \mu \sum_i |\theta_{i+1} - \theta_i|.$$

The ℓ_1 penalty on the increments $\theta_{i+1} - \theta_i$ favours piecewise constant solutions. Unfortunately, this twist on the standard lasso penalty renders coordinate descent inefficient. We can reformulate the problem as minimizing the criterion $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\theta}\|^2 + \mu\|\boldsymbol{\gamma}\|_1$ subject to the constraint $\boldsymbol{\gamma} = \boldsymbol{D\theta}$, where

$$d_{ij} = \begin{cases} 1 & j = i + 1 \\ -1 & j = i \\ 0 & \text{otherwise.} \end{cases}$$

In the augmented Lagrangian framework, updating $\boldsymbol{\theta}$ amounts to minimizing $1/2\ \|\boldsymbol{y} - \boldsymbol{\theta}\|^2 + \rho/2\ \|\boldsymbol{\gamma} - 1/\rho\boldsymbol{\lambda} - \boldsymbol{D\theta}\|^2$. It is straightforward to solve this least squares problem. Updating $\boldsymbol{\gamma}$ involves minimizing $\rho/2\ \|\boldsymbol{D\theta} - \boldsymbol{\gamma}\|^2 + \mu\|\boldsymbol{\gamma}\|_1$, which is a standard lasso problem. Thus, ADMM decouples the problematic linear transformation $\boldsymbol{D\theta}$ from the lasso penalty.

8 Algorithm Acceleration

Many MM and block descent algorithms converge very slowly. In partial compensation, the computational work per iteration may be light. Even so, diminishing the number of iterations until convergence by one or two orders of magnitude is an attractive proposition (Berlinet & Roland, 2007; Jamshidian & Jennrich, 1997; Kuroda & Sakakihara, 2006; Lange, 1995b; Roland & Varadhan, 2005; Zhou *et al.*, 2011). In this section, we discuss a generic method for accelerating a wide variety of algorithms (Zhou *et al.*, 2011). Consider a differentiable algorithm map $\boldsymbol{\theta}_{n+1} = A(\boldsymbol{\theta}_n)$ for optimizing an objective function $f(\boldsymbol{\theta})$, and suppose stationary points of $f(\boldsymbol{\theta})$ correspond to fixed points of $A(\boldsymbol{\theta})$. Equivalently, stationary points correspond

to roots of the equation $B(\boldsymbol{\theta}) = \boldsymbol{\theta} - A(\boldsymbol{\theta}) = \mathbf{0}$. Within this framework, it is natural to apply Newton's method

$$\begin{aligned}\boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n - dB(\boldsymbol{\theta}_n)^{-1} B(\boldsymbol{\theta}_n) \\ &= \boldsymbol{\theta}_n - [\boldsymbol{I} - dA(\boldsymbol{\theta}_n)]^{-1} B(\boldsymbol{\theta}_n) \end{aligned} \tag{9}$$

to find the root and accelerate the overall process. This is a realistic expectation because Newton's method converges at a quadratic rate in contrast to the linear rates of MM and block descent algorithms.

There are two principal impediments to implementing algorithm (9) in high dimensions. First, it appears to require evaluation and storage of the Jacobi matrix $dA(\boldsymbol{\theta})$, whose rows are the differentials of the components of $A(\boldsymbol{\theta})$. Second, it also appears to require inversion of the matrix $\boldsymbol{I} - dA(\boldsymbol{\theta})$. Both problems can be attacked by secant approximations. Close to the optimal point $\boldsymbol{\theta}_\infty$, the linear approximation

$$A \circ A(\boldsymbol{\theta}_n) - A(\boldsymbol{\theta}_n) \approx dA(\boldsymbol{\theta}_\infty)[A(\boldsymbol{\theta}_n) - \boldsymbol{\theta}_n]$$

is valid. This suggests that we take two ordinary steps and gather information in the process on the matrix $\boldsymbol{M} = A(\boldsymbol{\theta}_\infty)$. If we let $\boldsymbol{v}$ be the vector $A \circ A(\boldsymbol{\theta}_n) - A(\boldsymbol{\theta}_n)$ and $\boldsymbol{u}$ be the vector $A(\boldsymbol{\theta}_n) - \boldsymbol{\theta}_n$, then the secant condition reads $\boldsymbol{M}\boldsymbol{u} = \boldsymbol{v}$. In practice, it is advisable to exploit multiple secant conditions $\boldsymbol{M}\boldsymbol{u}_i = \boldsymbol{v}_i$ as long as their number does not exceed the number of parameters p. The secant conditions can be generated one per iteration over the current and previous $q - 1$ iterations. Let us represent the conditions collectively in the matrix form $\boldsymbol{M}\boldsymbol{U} = \boldsymbol{V}$ for $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q)$, and $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q)$.

The principle of parsimony suggests that we replace $\boldsymbol{M}$ by the smallest matrix satisfying the secant conditions. If we pose this problem concretely as minimizing the criterion $1/2 \, \|\boldsymbol{M}\|_F^2$ subject to the constraints $\boldsymbol{M}\boldsymbol{U} = \boldsymbol{V}$, then a straightforward exercise in Lagrange multipliers gives the solution $\boldsymbol{M} = \boldsymbol{V}(\boldsymbol{U}^t\boldsymbol{U})^{-1}\boldsymbol{U}^t$ (Lange, 2010). The matrix $\boldsymbol{M}$ has rank at most q, and the Sherman–Morrison formula yields that explicit inverse

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

Fortunately, it involves inverting just the $q \times q$ matrix $\boldsymbol{U}^t\boldsymbol{U} - \boldsymbol{U}^t\boldsymbol{V}$. Furthermore, the Newton update (9) boils down to

$$\begin{aligned}\boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n - [\boldsymbol{I} - \boldsymbol{V}(\boldsymbol{U}^t\boldsymbol{U})^{-1}\boldsymbol{U}^t]^{-1}[\boldsymbol{\theta}_n - A(\boldsymbol{\theta}_n)] \\ &= A(\boldsymbol{\theta}_n) - \boldsymbol{V}(\boldsymbol{U}^t\boldsymbol{U} - \boldsymbol{U}^t\boldsymbol{V})^{-1}\boldsymbol{U}^t[\boldsymbol{\theta}_n - A(\boldsymbol{\theta}_n)].\end{aligned}$$

The advantages of this procedure include the following: (a) it avoids large matrix inverses, (b) it relies on matrix times vector multiplication rather than matrix times matrix multiplication, (c) it requires only storage of the small matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and (d) it respects linear parameter constraints. Non-negativity constraints may be violated. The number of secants q should be fixed in advance, say between 1 and 15, and the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ should be updated by substituting the latest secant pair generated for the earliest secant pair retained. If an accelerated step fails the descent test, then one can revert to the ordinary MM or block descent step.

Acceleration of non-smooth algorithms is more problematic (Hiriart-Urruty & Lemarechal, 2001). For gradient descent and its generalizations (Combettes & Wajs, 2005) to non-smooth problems, Nesterov (2007) has suggested a potent acceleration. As noted by Beck &

Teboulle (2009), the accelerated iterates in ordinary gradient descent depend on an intermediate scalar t_n and an intermediate vector $\boldsymbol{\varphi}$ according to the formulas

$$\begin{aligned} t_{n+1} &= \frac{1+\sqrt{1+4t_n^2}}{2} \\ \boldsymbol{\varphi} &= \boldsymbol{\theta}_n + \frac{t_n - 1}{t_{n+1}}(\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n-1}) \\ \boldsymbol{\theta}_{n+1} &= \boldsymbol{\varphi} - \frac{1}{L}\nabla f(\boldsymbol{\varphi}) \end{aligned}$$

with initial values $t_1 = 1$ and $\boldsymbol{\varphi} = \boldsymbol{\theta}_0$. In other words, instead of taking a steepest descent step from the current iterate, one takes a steepest descent step from the extrapolated point $\boldsymbol{\varphi}$, which depends on both the current iterate $\boldsymbol{\theta}_n$ and the previous iterate $\boldsymbol{\theta}_{n-1}$. This mysterious extrapolation algorithm can yield impressive speedups for essentially the same computational cost per iteration as gradient descent.

9 Discussion

The fault lines in optimization separate smooth from non-smooth problems, unconstrained from constrained problems, and small-scale from large-scale problems. Smooth, unconstrained, and small-scale problems are easy to solve. Mathematical scientists are beginning to tackle non-smooth, constrained, large-scale problems at the opposite end of the difficulty spectrum. The most spectacular successes usually rely on convexity. We can expect further progress because some of the best minds in applied mathematics, computer science, and statistics have taken up the challenge. What is unlikely to occur is the discovery of a universally valid algorithm. Optimization is apt to remain as much art as science for a long time to come.

We have emphasized a few key ideas in this survey. Our examples demonstrate some of the possibilities for mixing and matching the different algorithm themes. Although we cannot predict the future of computational statistics with any certainty, the key ideas mentioned here will not disappear. For instance, penalization is here to stay, the descent property of an algorithm is always desirable, and quadratic approximation will always be superior to linear approximation for smooth functions. As computing devices hit physical constraints, the importance of parallel algorithms will also likely increase. This argues that block descent and parameter-separated MM algorithms will play a larger role in the future (Zhou *et al.*, 2010). Although we have de-emphasized convex calculus, readers who want to devise their own algorithms are well advised to learn this inherently subtle subject. There is a difference, after all, between principled algorithms and ad hoc procedures.

Acknowledgement

This research was supported in part by USPHS grants HG006139 and GM53275.

References

ACM SIGKDD and Netflix. (2007). *Proceedings of KDD Cup and Workshop*. Available online http://www.cs.uic.edu/liub/Net-flix-KDD-Cup-2007.html.

Beck, A. & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, **2**, 183–202.

Berlinet, A. & Roland, C. (2007). Acceleration schemes with application to the EM algorithm. *Comp. Statist. Data Anal.*, **51**, 3689–3702.
Bien, J. & Tibshirani, R.J. (2011). Sparse estimation of a covariance matrix. *Biometrika*, **98**(4), 807–820.
Böhning, D. & Lindsay, B.G. (1988). Monotonicity of quadratic approximation algorithms. *Ann. Instit. Stat. Math.*, **40**, 641–663.
Borwein, J.M. & Lewis, A.S. (2000). *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer.
Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, **3**(1), 1–122.
Bradley, E.L. (1973). The equivalence of maximum likelihood and weighted least squares estimates in the exponential family. *J. Amer. Statist. Assoc.*, **68**, 199–200.
Cai, J.-F., Candés, E.J. & Shen, Z. (2008). A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, **20**, 1956–1982.
Candés, E.J. & Tao, T. (2009). The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, **56**, 2053–2080.
Charnes, A., Frome, E.L. & Yu, P.L. (1976). The equivalence of generalized least squares and maximum likelihood in the exponential family. *J. Amer. Stat. Assoc.*, **71**, 169–171.
Chen, C., He, B. & Yuan, X. (2012). Matrix completion via an alternating direction method. *IMA J. Numer. Anal.*, **32**, 227–245.
Chen, H.F. (2002). *Stochastic Approximation and its Applications*. Dordrecht: Kluwer.
Chen, S.S., Donoho, D.L. & Saunders, M.A. (1998). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, **20**, 33–61.
Chi, E.C., Zhou, H., Ortega Del Vecchyo, D. & Lange, K. (2013). Genotype imputation via matrix completion. *Genome Res.*, **23**, 509–518.
Claerbout, J. & Muir, F. (1973). Robust modeling with erratic data. *Geophys.*, **38**, 826–844.
Combettes, P. & Wajs, V. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, **4**, 1168–1200.
Conn, A.R., Gould, N.I.M. & Toint, P.L. (1991). Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Math. Prog.*, **50**, 177–195.
Davidon, W.C. (1959). Variable metric methods for minimization. AEC Research and Development Report ANL–5990, Argonne National Laboratory. USA.
de Leeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*, Eds. Barra, J.R., Brodeau, F., Romier, G. & Van Cutsem, B., pp. 133–146. Amsterdam: North Holland Publishing Company.
de Leeuw, J. (1994). Block relaxation algorithms in statistics. In *Information Systems and Data Analysis*, Eds. Bock, H.H., Lenski, W. & Richter, M.M., pp. 308–325. New York: Springer.
Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Roy. Stat. Soc. B*, **39**, 1–38.
Dennis, J.E. Jr & Schnabel, R.B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia: SIAM.
Ding, C., Li, T. & Jordan, M.I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**, 45–55.
Dobson, A.J. (1990). *An Introduction to Generalized Linear Models*. London: Chapman & Hall.
Donoho, D. & Johnstone, I. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.
Duchi, J., Shalev-Shwartz, S., Singer, Y. & Chandra, T. (2008). Efficient projections onto the l_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pp. 272–279. New York: ACM.
Fortin, M. & Glowinski, R. (1983). Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems. *J. Appl. Math. Mech.*, **65**, 622–622.
Friedman, J., Hastie, T. & Tibshirani, R. (2007). Pathwise coordinate optimization. *Ann. Appl. Stat.*, **1**, 302–332.
Gabay, D. & Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comp. Math. Appl.*, **2**, 17–40.
Gifi, A. (1990). *Nonlinear Multivariate Analysis*. Hoboken, NJ: Wiley.
Glowinski, R. & Marrocco, A. (1975). Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de dirichlet nonlineaires. *Rev. Francaise dAut. Inf. Rech. Oper.*, **2**, 41–76.
Goldstein, A.A. (1964). Convex programming in Hilbert space. *Bull. Amer. Math. Soc.*, **70**, 709–710.
Green, P.J. (1984). Iteratively reweighted least squares for maximum likelihood estimation and some robust and resistant alternatives (with discussion). *J. Roy. Stat. Soc. B*, **46**, 149–192.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,* 2nd ed. New York: Springer.
Hestenes, M.R. (1969). Multiplier and gradient methods. *J. Optim. Theory Appl.*, **4**, 303–320.
Hiriart-Urruty, J.B. & Lemarechal, C. (1996). *Convex Analysis and Minimization Algorithms: Part 1: Fundamentals.* New York: Springer.
Hiriart-Urruty, J.B. & Lemarechal, C. (2001). *Convex Analysis and Minimization Algorithms: Part 2: Advanced Theory and Bundle Methods.* New York: Springer.
Hunter, D.R. & Lange, K. (2004). A tutorial on MM algorithms. *Amer. Statist.*, **58**, 30–37.
Jamshidian, M. & Jennrich, R.I. (1997). Quasi-Newton acceleration of the EM algorithm. *J. Roy. Stat. Soc. B*, **59**, 569–587.
Jank, W. (2006). Implementing and diagnosing the stochastic approximation EM algorithm. *J. Comput. Graph. Statist.*, **15**, 803–829.
Jennrich, R.I. & Moore, R.H. (1975). Maximum likelihood estimation by means of nonlinear least squares. In *Proceedings of the Statistical Computing Section: Amer. Stat. Assoc.*, Atlanta, Georgia, pp. 57–65.
Khalfan, H.F., Byrd, R.H. & Schnabel, R.B. (1993). A theoretical and experimental study of the symmetric rank-one update. *SIAM J. Optim.*, **3**, 1–24.
Kiefer, J. & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.*, **23**, 462–466.
Kruskal, J.B. (1965). Analysis of factorial experiments by estimating monotone transformations of the data. *J. Roy. Stat. Soc. B*, **27**, 251–263.
Kuroda, M. & Sakakihara, M. (2006). Accelerating the convergence of the EM algorithm using the vector epsilon algorithm. *Comput. Statist. Data Anal.*, **51**, 1549–1561.
Kushner, H.J. & Yin, G.G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications.* New York: Springer.
Lange, K. (1995). A gradient algorithm locally equivalent to the EM algorithm. *J. Roy. Stat. Soc. B*, **57**, 425–437.
Lange, K. (1995). A quasi-Newton acceleration of the EM algorithm. *Statist. Sinica*, **5**, 1–18.
Lange, K. (2010). *Numerical Analysis for Statisticians,* 2nd ed. New York: Springer.
Lange, K. (2012). *Optimization,* 2nd ed. New York: Springer.
Lange, K., Hunter, D.R. & Yang, I. (2000). Optimization transfer using surrogate objective functions (with discussion). *J. Comput. Graph. Statist.*, **9**, 1–59.
Lee, D.D. & Seung, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**, 788–791.
Levitin, E.S. & Polyak, B.T. (1966). Constrained minimization problems. *USSR Comput. Math and Math Physics*, **6**, 1–50.
Mazumder, R., Hastie, T. & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, **11**, 2287–2322.
McLachlan, G.J. & Krishnan, T. (2008). *The EM Algorithm and Extensions,* 2nd ed. Hoboken, NJ: Wiley.
Meinshausen, N. & Bühlmann, P. (2010). Stability selection. *J. Roy. Stat. Soc. B*, **72**, 417–473.
Meng, X.-L. & Rubin, D.B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, **80**, 267–278.
Michelot, C. (1986). A finite algorithm for finding the projection of a point onto the canonical simplex in R^n. *J. Optim. Theory Appl.*, **50**, 195–200.
Nelder, J.A. & Wedderburn, R.W.M. (1972). Generalized linear models. *J. Roy. Stat. Soc. A*, **135**, 370–384.
Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. CORE Discussion Papers.
Nocedal, J. & Wright, S. (2006). *Numerical Optimization,* 2nd ed. New York: Springer.
Ortega, J.M. & Rheinboldt, W.C. (1970). *Iterative Solutions of Nonlinear Equations in Several Variables.* New York: Academic.
Osborne, M.R. (1992). Fisher's method of scoring. *Int. Stat. Rev.*, **60**, 99–117.
Paatero, P. & Tapper, U. (1994). Positive matrix factorization: a non-negative factor model with optimal utilization of error. *Environmetrics*, **5**, 111–126.
Park, M.Y. & Hastie, T. (2007). ℓ_1-regularization path algorithm for generalized linear models. *J. Roy. Stat. Soc. B*, **69**, 659–677.
Powell, M.J.D. (1969). A method for nonlinear constraints in minimization problems. In *Optimization*, Ed. Fletcher, R., pp. 283–298. New York: Academic Press.
Qin, Z. & Goldfarb, D. (2012). Structured sparsity via alternating direction methods. *J. Mach. Learn. Res.*, **98888**, 1435–1468.
Ranola, J.M., Ahn, S., Sehl, M.E., Smith, D.J. & Lange, K. (2010). A Poisson model for random multigraphs. *Bioinformatics*, **26**, 2004–2011.
Rao, C.R. (1973). *Linear Statistical Inference and its Applications,* 2nd ed. Hoboken, NJ: Wiley.

Richard, E., Savalle, P.-A. & Vayatis, N. (2012). Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pp. 1351–1358. Edinburgh, Scotland, UK.

Robert, C. & Casella, G. (2004). *Monte Carlo Statistical Methods.* New York: Springer.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Ann. Math. Stat.*, **22**, 400–407.

Rockafellar, R.T. (1973). The multiplier method of Hestenes and Powell applied to convex programming. *J. Optim. Theory Appl.*, **12**, 555–562.

Roland, C. & Varadhan, R. (2005). New iterative schemes for nonlinear fixed point problems, with applications to problems with bifurcations and incomplete-data problems. *Appl. Numer. Math.*, **55**, 215–226.

Ruszczyński, A. (2006). *Nonlinear Optimization.* Princeton, NJ: Princeton University Press.

Santosa, F. & Symes, W.W. (1986). Linear inversion of band-limited reflection seimograms. *SIAM J. Sci. Stat. Comput.*, **7**, 1307–1330.

Sha, F., Saul, L.K. & Lee, D.D. (2003). Multiplicative updates for nonnegative quadratic programming in support vector machines. In *Advances in Neural Information Processing Systems*, Vol. 15, Eds. Becker, S., Thrun, S. & Obermayer, K., pp. 1065–1073. Cambridge, MA: MIT Press.

Strang, G. & Borre, K. (2012). *Algorithms for Global Positioning.* Wellesley, MA: Wellesley-Cambridge Press.

Taylor, H., Banks, S.C. & McCoy, J.F. (1979). Deconvolution with the ℓ_1 norm. *Geophys.*, **44**, 39–52.

Teo, C.H., Vishwanthan, S., Smola, A.J. & Le, Q.V. (2010). Bundle methods for regularized risk minimization. *J. Mach. Learn. Res.*, **11**, 311–365.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc., Ser. B*, **58**, 267–28.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. B*, **67**, 91–108.

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

Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E.M. & Lange, K. (2009). Genomewide association analysis by lasso penalized logistic regression. *Bioinformatics*, **25**, 714–721.

Wu, T.T. & Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, **2**, 224–244.

Wu, T.T. & Lange, K. (2010). The MM alternative to EM. *Stat. Sci.*, **25**, 492–505.

Xue, L., Ma, S. & Zou, H. (2012). Positive definite ℓ_1 penalized estimation of large covariance matrices. *JASA*, **107**, 1480–1491.

Zhou, H. & Zhang, Y. (2012). EM vs MM: a case study. *Comp. Stat. Data Anal.*, **56**, 3909–3920.

Zhou, H., Alexander, D.H. & Lange, K. (2011). A quasi-Newton acceleration for high-dimensional optimization algorithms. *Statist. Comput.*, **21**, 261–273.

Zhou, H. & Lange, K. (2010). MM algorithms for some discrete multivariate distributions. *J. Comput. Graph. Statist.*, **19**, 645–665.

Zhou, H., Lange, K. & Suchard, M.A. (2010). Graphics processing units and high-dimensional optimization. *Stat. Sci.*, **25**, 311–324.

Zou, H. (2006). The adaptive lasso and its oracle properties. *JASA*, **101**, 1418–1429.

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc. B*, **67**, 301–320.

[*Received September 2012, accepted April 2013*]