PNAS INAUGURAL ARTICLE | APPLIED MATHEMATICS

MM optimization: Proximal distance algorithms, path following, and trust regions

Alfonso Landeros[a], Jason Xu[b], and Kenneth Lange[a,c,d,1]

This contribution is part of the special series of Inaugural Articles by members of the National Academy of Sciences elected in 2021.
Contributed by Kenneth Lange; received March 3, 2023; accepted May 9, 2023; reviewed by David Hunter and Ravi Varadhan

We briefly review the majorization–minimization (MM) principle and elaborate on the closely related notion of proximal distance algorithms, a generic approach for solving constrained optimization problems via quadratic penalties. We illustrate how the MM and proximal distance principles apply to a variety of problems from statistics, finance, and nonlinear optimization. Drawing from our selected examples, we also sketch a few ideas pertinent to the acceleration of MM algorithms: a) structuring updates around efficient matrix decompositions, b) path following in proximal distance iteration, and c) cubic majorization and its connections to trust region methods. These ideas are put to the test on several numerical examples, but for the sake of brevity, we omit detailed comparisons to competing methods. The current article, which is a mix of review and current contributions, celebrates the MM principle as a powerful framework for designing optimization algorithms and reinterpreting existing ones.

optimization | statistics | data science | computation

Significance

Optimization methods are essential to applied mathematics, statistics, and machine learning. By converting a difficult high-dimensional optimization problem into a sequence of simpler ones, the majorization–minimization (MM) principle is a versatile tool for designing novel algorithms. Clever application of inequalities from analysis allows practitioners to design surrogate functions that are easy to optimize and imbue an MM algorithm with desirable characteristics. For example, an MM algorithm can separate parameters, allowing for parallel updates, or reduce an update step to solving a system of linear equations. Distance majorization further extends the reach of MM to constrained optimization problems.

This article discusses recent developments and presents ideas for designing iterative optimization algorithms from the majorization–minimization (MM) perspective. The MM principle is now well entrenched in high-dimensional optimization (1–4). Its long and fascinating history is briefly covered in the preface to the book (5). MM operates by converting a difficult optimization problem into a sequence of simpler ones. Designing an algorithm from the MM perspective can: a) avoid large matrix inversions, b) linearize an optimization problem, c) separate the variables of a problem, d) deal with equality and inequality constraints gracefully, and e) turn a nondifferentiable problem into a smooth problem. All MM algorithms are descent algorithms in minimization and ascent algorithms in maximization. They therefore tend to make steady, stable progress with fast updates. Although MM algorithms can take hundreds of iterations to converge, excess iterations can be trimmed by well-studied acceleration techniques such as SQUAREM and quasi-Newton schemes (6–10). The MM principle also plays well with other optimization methods such as block optimization, gradient descent, alternating least squares, and local quadratic approximation.

In minimization, MM iteratively substitutes a surrogate function that majorizes the original loss $f(\boldsymbol{x})$ around the current iterate $\boldsymbol{x}_n$. The surrogate function $g(\boldsymbol{x} \mid \boldsymbol{x}_n)$ majorizes the loss at iteration n provided it satisfies the tangency condition $g(\boldsymbol{x}_n \mid \boldsymbol{x}_n) = f(\boldsymbol{x}_n)$ and the domination condition $g(\boldsymbol{x} \mid \boldsymbol{x}_n) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x}$. The surrogate balances two goals, hugging the objective tightly and simplifying minimization. Minimizing this surrogate produces the next iterate $\boldsymbol{x}_{n+1}$ and drives the objective downhill owing to conditions

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

In maximization, the surrogate minorizes the objective and instead must be maximized. The tangency condition remains the same, but the domination condition $g(\boldsymbol{x} \mid \boldsymbol{x}_n) \leq f(\boldsymbol{x})$ is now reversed. The celebrated EM (expectation–maximization) principle for maximum likelihood estimation with missing data (11) is a special case of minorization–maximization. In the EM setting, Jensen's inequality supplies the surrogate as the expectation of the complete data log-likelihood conditional on the observed data. Majorization in the MM sense should not be confused with the notion of majorization in the theory of inequalities (12). Both notions rely heavily on convexity, but the resemblance seems to end there.

In the current paper, a focus is on constructing surrogates based on the idea of distance majorization (13). The proximal distance principle (14) addresses constrained

Author affiliations: [a]Department of Computational Medicine, University of California, Los Angeles, CA 90095; [b]Department of Statistical Science, Duke University, Durham, NC 27708; [c]Department of Human Genetics, University of California, Los Angeles, CA 90095; and [d]Department of Statistics, University of California, Los Angeles, CA 90095

Author contributions: A.L., J.X., and K.L. designed research; A.L., J.X., and K.L. performed research; A.L. wrote code; and A.L., J.X., and K.L. wrote the paper.

Reviewers: D.H., Pennsylvania State University; and R.V., Johns Hopkins University.

The authors declare no competing interest.

[1]To whom correspondence may be addressed. Email: klange@ucla.edu.

Published June 20, 2023.

minimization of a loss $f(\boldsymbol{x})$ subject to $\boldsymbol{x}$ belonging to a closed set C. Constraints such as nonnegativity, sparsity, and low matrix rank bring many applications into the MM fold (15–18). The proximal distance principle leverages Courant's penalty method (19, 20) and Euclidean set projections, which are often explicitly known or reduce to fast algorithms (21, 22). Courant's penalty requires adding a nonnegative penalty vanishing on the constraint set. In our context, this agenda is carried out by minimizing

$$h_\rho(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{\rho}{2}\operatorname{dist}(\boldsymbol{x}, C)^2, \quad \rho > 0. \tag{1}$$

One then follows its solution vector $\boldsymbol{x}_\rho$ as $\rho \to \infty$ to the solution of the constrained problem. By design, the penalty $\frac{\rho}{2}\operatorname{dist}(\boldsymbol{x}, C)^2 = 0$ precisely when $\boldsymbol{x} \in C$. For fixed ρ, the distance majorization

$$\operatorname{dist}(\boldsymbol{x}, C)^2 = \inf_{\boldsymbol{y}\in C} \|\boldsymbol{x}-\boldsymbol{y}\|^2 \quad \leq \quad \|\boldsymbol{x} - P_C(\boldsymbol{x}_n)\|^2,$$

pertinent to the current iterate $\boldsymbol{x}_n$ leads to a simple update. Here, $P_C(\boldsymbol{z})$ denotes the Euclidean projection of $\boldsymbol{z}$ onto C. The tangency condition $\inf_{\boldsymbol{y}\in C}\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x} - P_C(\boldsymbol{x})\|$ is true by definition. To construct the next iterate $\boldsymbol{x}_{n+1}$, the surrogate function

$$g_\rho(\boldsymbol{x} \mid \boldsymbol{x}_n) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - P_C(\boldsymbol{x}_n)\|^2,$$

is minimized via the proximal map $\operatorname{prox}_{\rho^{-1}f}[P_C(\boldsymbol{x}_n)]$ (21, 22). A wide variety of functions generate readily computable proximal maps. For ρ fixed the MM principle guarantees that minimizing $g_\rho(\boldsymbol{x} \mid \boldsymbol{x}_n)$ reduces the penalized loss [**1**].

In practice, we solve a sequence of subproblems parameterized by an annealing schedule ρ_n. The idea is to choose ρ_0 small (say 1) and to increase ρ_n by gradual multiplicative increments to a large maximum value (say 10^6 to 10^8). Unfortunately, the quality of the solution to the inner problem of minimizing the penalized loss for fixed ρ_n affects the quality of the outer solution when ρ_n reaches its maximum. The gradient condition $\|\nabla h_\rho(\boldsymbol{x})\| \leq \delta$ for δ small is the safest option for declaring convergence in the inner iterations. We also monitor the closeness of the solution vector $\boldsymbol{x}_\rho$ to the constraint set in the outer iterations.

Our recent paper (23) further generalizes the proximal distance method to fusion constraints $\boldsymbol{D}\boldsymbol{x} \in C$ involving linear transformations $\boldsymbol{D}\boldsymbol{x}$ of $\boldsymbol{x}$. Now the surrogate

$$g_\rho(\boldsymbol{x} \mid \boldsymbol{x}_n) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - P_C(\boldsymbol{D}\boldsymbol{x}_n)\|^2. \tag{2}$$

relies on a nonspherical quadratic penalty, and its minimum falls outside the realm of proximal maps. In many problems, the loss is a convex quadratic

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_n) + \nabla f(\boldsymbol{x}_n)^\top(\boldsymbol{x}-\boldsymbol{x}_n) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_n)^\top\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_n), \tag{3}$$

with a constant Hessian $\boldsymbol{H}$. For nonquadratic losses, the right-hand side of Eq. **3** often furnishes a quadratic upper-bound (24) majorization of $f(\boldsymbol{x})$. Hence, minimization of the summed surrogate

$$f(\boldsymbol{x}_n) + \nabla f(\boldsymbol{x}_n)^\top(\boldsymbol{x}-\boldsymbol{x}_n) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_n)^\top\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_n) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - P_C(\boldsymbol{D}\boldsymbol{x}_n)\|^2,$$

is guaranteed to drive the penalized loss $h_\rho(\boldsymbol{x})$ downhill. In practice, the descent property usually holds empirically even when the loss is merely well-approximated by the convex quadratic [**3**]. We will pay special attention to surrogates [**2**] and [**3**] because they lead to straightforward MM updates.

In the remaining sections, we summarize several broad application areas that benefit from the MM perspective. Then, we highlight three computational improvements to MM: a) Since a proximal distance algorithm solves a sequence of problems as the penalty constants $\rho_n \to \infty$, structuring repeated computations around key intermediates, such as spectral decompositions, can save time. Here, the quadratic expansion [**3**] plays a prominent role. b) Based on implicit differentiation, proximal distance algorithms can benefit from path following as $\rho \to \infty$. In practice, this amounts to projecting the solution at ρ_n to the solution at ρ_{n+1}. The proximal distance algorithm then corrects this accurate guess. c) The interplay between trust regions and local majorization, two prominent tools in optimization, has been largely overlooked. Local majorization is valuable because many objective functions $f(\boldsymbol{x})$ exhibit cubic or stronger coerciveness as $\|\boldsymbol{x}\| \to \infty$ and lack quadratic majorizers. We show how to replace a cubic majorization by a local quadratic majorization, which is then easily minimized. This tactic expands the scope of problems that can be tackled by MM.

Here are the notational conventions used throughout this article. All vectors and matrices appear in boldface. All entries of the vector $\mathbf{0}$ equal 0; $\boldsymbol{I}$ is an identity matrix. The $^\top$ superscript indicates a vector or matrix transpose. The Euclidean norm of a vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|$, and the spectral and Frobenius norms of a matrix $\boldsymbol{M}$ are denoted by $\|\boldsymbol{M}\|$ and $\|\boldsymbol{M}\|_F$, respectively. For a smooth real-valued function $f(\boldsymbol{x})$, we write its gradient (column vector of partial derivatives) as $\nabla f(\boldsymbol{x})$, its first differential (row vector of partial derivatives) as $df(\boldsymbol{x}) = \nabla f(\boldsymbol{x})^\top$, and its second differential (Hessian matrix) as $d^2 f(\boldsymbol{x})$. The symbol $\otimes$ signals the Kronecker product of two matrices. The vec operation stacks the columns of $\boldsymbol{M}$ into a single column vector $\operatorname{vec}(\boldsymbol{M})$. A vector sequence $\boldsymbol{x}_n$ has ith component x_{ni}. Finally, $B_r(\boldsymbol{x}) = \{\boldsymbol{y} : \|\boldsymbol{y}-\boldsymbol{x}\| \leq r\}$ denotes the closed ball of radius r and center $\boldsymbol{x}$.

1. Sample MM Algorithms

Let us consider a few examples of MM and proximal distance algorithms.

Example 1.1: Constrained Least Squares.

The now-classic fused lasso problem involves minimizing the criterion $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ subject to the constraint $\|\boldsymbol{D}\boldsymbol{\beta}\|_1 \leq \epsilon$ (25), where $\boldsymbol{D}$ is the difference matrix with first row $(1, -1, \boldsymbol{0})$, second row $(0, 1, -1, \boldsymbol{0})$, and so forth. Let B be the ℓ_1 ball $\{\boldsymbol{y} : \|\boldsymbol{y}\|_1 \leq \epsilon\}$. The proximal distance algorithm minimizes the least squares surrogate

$$g_\rho(\boldsymbol{\beta} \mid \boldsymbol{\beta}_n) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{\beta} - P_B(\boldsymbol{D}\boldsymbol{\beta}_n)\|^2 = \frac{1}{2}\left\|\begin{pmatrix}\boldsymbol{X}\\ \sqrt{\rho}\boldsymbol{D}\end{pmatrix}\boldsymbol{\beta} - \begin{pmatrix}\boldsymbol{b}\\ \sqrt{\rho}P_B(\boldsymbol{D}\boldsymbol{\beta}_n)\end{pmatrix}\right\|^2,$$

to update $\boldsymbol{\beta}$. The case $\boldsymbol{D} = \boldsymbol{I}$, corresponding to the classic lasso, is included. To handle other constraints such as nonnegative least squares or sparse regression, all that changes is the projection operator. In the former case, one projects onto the nonnegative orthant $\mathbb{R}^d_+$, and in the latter case one projects onto the nonconvex sparsity set $S_k = \{\boldsymbol{x} : \|\boldsymbol{x}\|_0 \leq k\}$ for some sparsity level k (21, 22). These ideas are also pertinent to maximum likelihood estimation under generalized linear models (15), where the loss is well approximated by a sum of squares criterion. □

Example 1.2: Linear and Semidefinite Programming.

Many programming problems involve minimizing a linear function $\boldsymbol{c}^\top\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\boldsymbol{x} \in S$, where S is a closed set. For instance in linear programming, $S = \mathbb{R}^p_+$. One can phrase the general case as a proximal distance problem with objective $\boldsymbol{c}^\top\boldsymbol{x} + \frac{\rho}{2}\,\mathrm{dist}(\boldsymbol{x}, S)^2$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Let us take $\boldsymbol{c}^\top\boldsymbol{x} + \frac{\rho}{2}\|\boldsymbol{x} - P_S(\boldsymbol{x}_n)\|^2$ as the surrogate function subject to the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Minimization of the surrogate via Lagrange multipliers generates the stationary condition

$$\begin{aligned}\boldsymbol{0} &= \boldsymbol{c} + \rho[\boldsymbol{x} - P_S(\boldsymbol{x}_n)] + \boldsymbol{A}^\top\boldsymbol{\lambda}\\ \boldsymbol{0} &= \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b},\end{aligned} \qquad [4]$$

for the Lagrangian. This translates into the single equation

$$\begin{pmatrix}\rho\boldsymbol{I} & \boldsymbol{A}^\top\\ \boldsymbol{A} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{\lambda}\end{pmatrix} = \begin{bmatrix}\rho P_S(\boldsymbol{x}_n) - \boldsymbol{c}\\ \boldsymbol{b}\end{bmatrix},$$

which can be solved for $\boldsymbol{x}_{n+1}$ and the Lagrange multiplier $\boldsymbol{\lambda}$ simultaneously.

Alternatively, we can multiply Eq. **4** by $\boldsymbol{A}$, substitute $\boldsymbol{b}$ for $\boldsymbol{A}\boldsymbol{x}$, and solve for $\boldsymbol{\lambda}$ in the form

$$\boldsymbol{\lambda} = (\boldsymbol{A}\boldsymbol{A}^\top)^{-1}[\rho\boldsymbol{A}P_S(\boldsymbol{x}_n) - \rho\boldsymbol{b} - \boldsymbol{A}\boldsymbol{c}],$$

and then for $\boldsymbol{x}$ as

$$\begin{aligned}\boldsymbol{x}_{n+1} &= \rho^{-1}[\rho P_S(\boldsymbol{x}_n) - \boldsymbol{c} - \boldsymbol{A}^\top\lambda] = P_S(\boldsymbol{x}_n) - \rho^{-1}\boldsymbol{c} - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}[\boldsymbol{A}P_S(\boldsymbol{x}_n) - \boldsymbol{b} - \rho^{-1}\boldsymbol{A}\boldsymbol{c}]\\ &= [\boldsymbol{I} - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}][P_S(\boldsymbol{x}_n) - \rho^{-1}\boldsymbol{c}] + \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{b}.\end{aligned}$$

All iterates $\boldsymbol{x}_n$ require the same pseudoinverse $\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}$, which only needs to be computed once and stored.

In semidefinite programming, we replace $\boldsymbol{x}$ and $\boldsymbol{c}$ by symmetric matrices $\boldsymbol{X}$ and $\boldsymbol{C}$. The linear function $\boldsymbol{c}^\top\boldsymbol{x}$ becomes $\mathrm{tr}(\boldsymbol{C}^\top\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{C})^\top\,\mathrm{vec}(\boldsymbol{X})$, and the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has rows $\mathrm{tr}(\boldsymbol{A}_i^\top\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{A}_i)^\top\,\mathrm{vec}(\boldsymbol{X}) = b_i$ for m symmetric matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$. The relevant constraint set S is now the set of positive semidefinite matrices. □

Example 1.3: Projection onto a Set Intersection

Euclidean projection minimizes the spherical quadratic $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$ subject to $\boldsymbol{x} \in C$ for C closed. If C is the intersection of closed sets D and E, then often the operator $P_{D\cap E}(\boldsymbol{y})$ cannot be easily expressed in terms of $P_D(\boldsymbol{Y})$ and $P_E(\boldsymbol{y})$. The iterative methods of alternating and averaged projections yield in the limit points in $D \cap E$. Dykstra's algorithm (26) supplies $P_{D\cap E}(\boldsymbol{y})$. The proximal distance principle offers another possibility for finding this projection. Consider the function $f_\rho(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \frac{\rho}{2}\,\mathrm{dist}(\boldsymbol{x}, E)^2$ for ρ large. If we majorize $f_\rho(\boldsymbol{x})$ by

$$g_\rho(\boldsymbol{x} \mid \boldsymbol{x}_n) = \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{x}\right\|^2 + \frac{\rho}{2}\|\boldsymbol{x} - P_E(\boldsymbol{x}_n)\|^2 = \frac{1+\rho}{2}\left\|\boldsymbol{x} - \frac{1}{1+\rho}\boldsymbol{y} - \frac{\rho}{1+\rho}P_E(\boldsymbol{x}_n)\right\|^2 + k_n,$$

where k_n is an irrelevant constant, then the iteration scheme

$$\boldsymbol{x}_{n+1} = P_D\left[\frac{1}{1+\rho}\boldsymbol{y} + \frac{\rho}{1+\rho}P_E(\boldsymbol{x}_n)\right], \qquad [5]$$

reduces $f_\rho(\boldsymbol{x})$ on the set D. As an example, D could be the unit simplex and E the sparsity set S_k where at most k components of $\boldsymbol{x}$ are nonzero. Because projection onto a closed convex set is nonexpansive, the map [**5**] is a contraction mapping with contraction constant $\frac{\rho}{1+\rho}$ when E is convex. The MM principle is also useful for projection onto Minkowski set sums (27) and split feasibility problems (16). □

Example 1.4: Nearest Sparse Covariance Matrix

Our recent article (17) deals with the problem of estimating a covariance matrix Σ subject to sparsity constraints (28). Given m i.i.d. normal deviates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \sim N_p(\boldsymbol{0}, \Sigma)$, the rescaled and penalized negative log-likelihood is

$$h_\rho(\Sigma) = \frac{1}{2}\ln\det\Sigma + \frac{1}{2}\mathrm{tr}(\Sigma^{-1}\boldsymbol{S}) + \frac{\rho}{2}\mathrm{dist}(\Sigma, C_k)^2, \quad [6]$$

where $C_k = \left\{\Sigma \in \mathbb{R}^{p\times p} : \Sigma = \Sigma^\top, \|\Sigma\|_0 \le 2k\right\}$ denotes the sparsity set with at most k nonzero strictly upper triangular entries and $\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^m \boldsymbol{x}_i\boldsymbol{x}_i^\top$ denotes the sample covariance matrix. The loss $f(\Sigma) = \frac{1}{2}\ln\det\Sigma + \frac{1}{2}\mathrm{tr}(\Sigma^{-1}\boldsymbol{S})$ is not quadratic but can be locally approximated by a quadratic. Matrix calculus shows that the Hessian $d^2f(\Sigma)$ generates the quadratic form

$$\boldsymbol{V}^\top d^2f(\Sigma)\boldsymbol{V} := -\mathrm{tr}(\Sigma^{-1}\boldsymbol{V}\Sigma^{-1}\boldsymbol{V}) + 2\,\mathrm{tr}(\Sigma^{-1}\boldsymbol{V}\Sigma^{-1}\boldsymbol{V}\Sigma^{-1}\boldsymbol{S}),$$

on matrices $\boldsymbol{V}$. Computationally, it is advantageous to imitate Fisher scoring and replace $\boldsymbol{S}$ by its expected value $\mathbb{E}(\boldsymbol{S}) = \Sigma$. The quadratic form then simplifies to

$$\boldsymbol{V}^\top d^2f(\Sigma)\boldsymbol{V} \approx \mathrm{tr}(\Sigma^{-1}\boldsymbol{V}\Sigma^{-1}\boldsymbol{V}),$$

which is guaranteed to be positive definite. After distance majorization, this maneuver yields the approximate but tractable quadratic surrogate

$$\begin{aligned} q_\rho(\Sigma \mid \Sigma_n) = {} & f(\Sigma_n) + \mathrm{tr}[\Sigma_n^{-1}(\Sigma - \Sigma_n)] - \mathrm{tr}[\Sigma_n^{-1}\boldsymbol{S}\Sigma_n^{-1}(\Sigma - \Sigma_n)] + \frac{1}{2}\mathrm{tr}[\Sigma_n^{-1}(\Sigma - \Sigma_n)\Sigma_n^{-1}(\Sigma - \Sigma_n)] \\ & + \frac{\rho}{2}\|\Sigma - P_{C_k}(\Sigma_n)\|_F^2, \end{aligned}$$

whose minimization almost always decreases $h_\rho(\Sigma)$.

Equating the gradient of $h_\rho(\Sigma)$ to $\boldsymbol{0}$ gives the stationary condition

$$\rho P_{C_k}(\Sigma_n) + \Sigma_n^{-1}\boldsymbol{S}\Sigma_n^{-1} = \rho\Sigma + \Sigma_n^{-1}\Sigma\Sigma_n^{-1}, \quad [7]$$

which after vectorization becomes

$$\mathrm{vec}(\boldsymbol{M}_n) = \rho\,\mathrm{vec}(\Sigma) + (\Sigma_n^{-1} \otimes \Sigma_n^{-1})\,\mathrm{vec}(\Sigma),$$

where we abbreviate $\boldsymbol{M}_n = \rho P_{C_k}(\Sigma_n) + \Sigma_n^{-1}\boldsymbol{S}\Sigma_n^{-1}$. The solution to this vector equation

$$\mathrm{vec}(\widehat{\Sigma}) = \left[\rho I_{p^2} + (\Sigma_n^{-1} \otimes \Sigma_n^{-1})\right]^{-1}\mathrm{vec}(\boldsymbol{M}_n), \quad [8]$$

can be reshaped to form the matrix minimizer $\widehat{\Sigma}$. This straightforward solution reveals the utility of quadratic majorization. Unfortunately, the solution requires inverting a $p^2 \times p^2$ matrix, which quickly becomes intractable as p grows. However, Eq. 7 assumes the general form $\boldsymbol{A}\Sigma + \Sigma\boldsymbol{B} = \boldsymbol{C}$ after multiplying both sides by Σ_n. This we recognize as a Sylvester equation in Σ, to which powerful numerical methods such as the Bartels–Stewart algorithm apply (29). The computational complexity declines to $O(p^3)$. These considerations demonstrate how careful decompositions can yield efficient updates. □

Example 1.5: Portfolio Optimization

Recently De Simone et al. devised an interior point method for the portfolio problem

$$\begin{aligned} \min_{\boldsymbol{W}} \quad & \frac{1}{2}\sum_{j=1}^m \boldsymbol{w}_j^\top \boldsymbol{C}_j\boldsymbol{w}_j + \tau_1\|\boldsymbol{w}_j\|_1 + \tau_2\sum_{j=1}^{m-1}\|\boldsymbol{w}_{j+1} - \boldsymbol{w}_j\|_1, \\ \text{s.t.} \quad & \boldsymbol{1}_s^\top\boldsymbol{w}_1 = \xi_{\text{init}}, \\ & \boldsymbol{1}_s^\top\boldsymbol{w}_{j+1} = (\boldsymbol{1}_s^\top + \boldsymbol{r}_j)^\top\boldsymbol{w}_j \text{ for } j = 1, 2, \ldots, m, \\ & (\boldsymbol{1}_s + \boldsymbol{r}_j)^\top\boldsymbol{w}_m = \xi_{\text{final}}, \end{aligned}$$

by rewriting the usual convex program as an equivalent quadratic program (30). The s-dimensional columns $\boldsymbol{w}_j$ denote wealth invested across s assets at time period j; $\boldsymbol{C}_j$ is a covariance matrix on returns $\boldsymbol{r}_j$ across the assets over the same time period. The absence of nonnegativity constraints on components of the $\boldsymbol{w}_j$ allows the model to accommodate short selling. The affine constraints reflect

rebalancing between periods to grow initial wealth ξ_{init} to a desired outcome ξ_{final}. Alternatively, the proximal distance principle suggests minimizing

$$h_\rho(\boldsymbol{W}) = \frac{1}{2}\sum_{j=1}^{m} \boldsymbol{w}_j^\top \boldsymbol{C}_j \boldsymbol{w}_j + \frac{\rho}{2}\sum_{k=1}^{3} \alpha_k \operatorname{dist}[\boldsymbol{D}_k \operatorname{vec}(\boldsymbol{W}), S_k]^2,$$

via the surrogate quadratic model

$$g_\rho(\boldsymbol{W} \mid \boldsymbol{W}_n) = \frac{1}{2}\sum_{j=1}^{m} \boldsymbol{w}_j^\top \boldsymbol{C}_j \boldsymbol{w}_j + \frac{\rho}{2}\sum_{k=1}^{3} \alpha_k \|P_{S_k}[\boldsymbol{D}_k \operatorname{vec}(\boldsymbol{W}_n)] - \boldsymbol{D}_k \operatorname{vec}(\boldsymbol{W})\|^2.$$

Here S_1 and S_2 indicate ℓ_1 balls with radii τ_1 and τ_2, respectively, and S_3 captures the affine constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ on wealth allocations $\boldsymbol{w}_j$ over time. The corresponding fusion matrices are $\boldsymbol{D}_1 = \boldsymbol{D}_3 = \boldsymbol{I}$ and $\boldsymbol{D}_2$, defined by its action $[\boldsymbol{D}_2 \operatorname{vec}(\boldsymbol{W})] = [\boldsymbol{w}_{j+1} - \boldsymbol{w}_j]$. The α_k are convex weights that reflect the relative importance of each distance penalty. The interpretation of the τ_i differs from the original formulation; these constants now impose explicit limits on wealth allocations and changes in allocations. □

The next two examples focus on unconstrained optimization. We will later study them in the context of a trust region method motivated from the MM perspective.

Example 1.6: Styblinski–Tang Function

The Styblinski–Tang function (31)

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{d} \left(x_i^4 - 16x_i^2 + 5x_i\right),$$

is difficult to minimize due to the presence of multiple extrema and the fact that its Hessian is singular at $x_i = \pm(8/3)^{1/2}$ for $i = 1, 2, \ldots, d$. It is often used as a benchmark for optimization techniques (32). The global minimum value $-39.1662\, d$ is attained when each $x_i \approx -2.9035340$. □

Example 1.7: Poisson Regression

In Poisson regression with inverse link $\mu(\boldsymbol{x}^\top\boldsymbol{\beta}) = E[y \mid \boldsymbol{x}]$, one maximizes the log-likelihood

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{m} y_i \ln \mu(\boldsymbol{x}_i^\top\boldsymbol{\beta}) - \ln y_i! - \mu(\boldsymbol{x}_i^\top\boldsymbol{\beta}),$$

where $\boldsymbol{y}$ is the response vector of length m, $\boldsymbol{x}_i$ is the predictor vector associated with response i, $\boldsymbol{\beta}$ is the regression coefficient vector, and $\mu(\boldsymbol{x}_i^\top\boldsymbol{\beta})$ is the mean of response i. The log-likelihood has score and expected information

$$\begin{aligned}\nabla\mathcal{L}(\boldsymbol{\beta}) &= \sum_{i=1}^{m} \mu'(\boldsymbol{x}_i^\top\boldsymbol{\beta})\left[\frac{y_i}{\mu(\boldsymbol{x}_i^\top\boldsymbol{\beta})} - 1\right]\boldsymbol{x}_i \\ J(\boldsymbol{\beta}) &= \sum_{i=1}^{m} \frac{\mu'(\boldsymbol{x}_i^\top\boldsymbol{\beta})^2}{\mu(\boldsymbol{x}_i^\top\boldsymbol{\beta})}\boldsymbol{x}_i\boldsymbol{x}_i^\top \quad = \quad \boldsymbol{X}^\top\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X},\end{aligned}$$

where $\boldsymbol{X}$ is the design matrix, and $\boldsymbol{W}$ is diagonal with ith diagonal entry $\frac{\mu'(\boldsymbol{x}_i^\top\boldsymbol{\beta})^2}{\mu(\boldsymbol{x}_i^\top\boldsymbol{\beta})}$. □

2. Efficient Proximal Distance Updates

As already mentioned, a proximal distance algorithm solves a sequence of unconstrained problems tagged by an increasing sequence of penalty constants ρ_n. Considerable computational efficiencies can be realized by saving key intermediates and structuring repeated computations accordingly. These techniques pay significant dividends in applications that rely on cross-validation to tune model hyperparameters.

For quadratic loss functions, the surrogate defined by Eqs. **2** and **3** satisfies the stationary condition

$$\boldsymbol{0} = \nabla f(\boldsymbol{x}_n) + (\boldsymbol{H} + \rho\boldsymbol{D}^\top\boldsymbol{D})(\boldsymbol{x} - \boldsymbol{x}_n) + \rho\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{x}_n - \boldsymbol{p}_n),$$

with solution

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - (\boldsymbol{H} + \rho\boldsymbol{D}^\top\boldsymbol{D})^{-1}[\nabla f(\boldsymbol{x}_n) + \rho\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{x}_n - \boldsymbol{p}_n)].$$

Here $\boldsymbol{H}$ is assumed positive semidefinite and $\boldsymbol{p}_n$ denotes the projection of $\boldsymbol{D}\boldsymbol{x}_n$ onto C. If either $\boldsymbol{H}$ or $\boldsymbol{D}$ is a multiple of the identity, then a single spectral decomposition renders the inverses $(\boldsymbol{H}+\rho\boldsymbol{D}^\top\boldsymbol{D})^{-1}$ trivial to compute across all values of ρ.

Until now, it has eluded us how to deal with the more general situation where neither $\boldsymbol{H}$ nor $\boldsymbol{D}$ is a multiple of the identity. We describe two viable strategies here. Both assume that $\boldsymbol{H}$ is positive definite and take its Cholesky decomposition $\boldsymbol{H}=\boldsymbol{L}\boldsymbol{L}^\top$. If the dimension p of $\boldsymbol{x}$ is low, then at each iteration one can efficiently promote the Cholesky decomposition $\boldsymbol{H}$ to the Cholesky decomposition of the low-rank perturbation $\boldsymbol{H}+\rho\boldsymbol{D}^\top\boldsymbol{D}$ (33). The second strategy extracts the spectral decomposition $\boldsymbol{O}\Sigma\boldsymbol{O}^\top$ of the matrix $\boldsymbol{L}^{-1}\boldsymbol{D}^\top\boldsymbol{D}(\boldsymbol{L}^{-1})^\top$. If we set $\boldsymbol{C}=(\boldsymbol{L}^{-1})^\top\boldsymbol{O}$, then the simultaneous diagonalization $\boldsymbol{C}^\top\boldsymbol{H}\boldsymbol{C}=\boldsymbol{I}_p$ and $\boldsymbol{C}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{C}=\Sigma$ of $\boldsymbol{H}$ and $\boldsymbol{D}^\top\boldsymbol{D}$ emerges. This allows us to write

$$(\boldsymbol{H}+\rho\boldsymbol{D}^\top\boldsymbol{D})^{-1}=\boldsymbol{C}(\boldsymbol{I}_p+\rho\Sigma)^{-1}\boldsymbol{C}^\top=(\boldsymbol{L}^{-1})^\top\boldsymbol{O}(\boldsymbol{I}_p+\rho\Sigma)^{-1}\boldsymbol{O}^\top\boldsymbol{L}^{-1}. \quad [9]$$

Under both strategies, the optimal $\boldsymbol{x}$ can now be found by a sequence of matrix-vector multiplications and forward-backward solves based on the Cholesky decomposition of $\boldsymbol{H}$. All of these operations share a computational complexity of $O(p^2)$. Of course, the middle matrix inverse in Eq. **9** is trivial because $\boldsymbol{I}_p+\rho\Sigma$ is diagonal. Again we stress that the same simultaneous diagonalization pertains to all ρ and only needs to be computed once and its components stored. Note that in least squares problems $\boldsymbol{H}=\boldsymbol{X}^\top\boldsymbol{X}$, so its Cholesky decomposition can be extracted from the QR decomposition of $\boldsymbol{X}$. Similarly, the spectral decomposition of $\boldsymbol{L}^{-1}\boldsymbol{D}^\top\boldsymbol{D}(\boldsymbol{L}^{-1})^\top$ can be extracted from the SVD of the matrix $\boldsymbol{L}^{-1}\boldsymbol{D}^\top=[\boldsymbol{D}(\boldsymbol{L}^{-1})^\top]^\top$.

Let us illustrate the second acceleration strategy on the portfolio optimization problem of Example 1.5 using S&P 500 data as prepared by Bruni et al. (34) with some assets dropped. The loss $f(\boldsymbol{W})=\sum_{j=1}^m\boldsymbol{w}_j^\top\boldsymbol{C}_j\boldsymbol{w}_j$ can be written $\operatorname{vec}(\boldsymbol{W})^\top\boldsymbol{C}\operatorname{vec}(\boldsymbol{W})$ with $\boldsymbol{C}$ block diagonal. Each block in turn is positive definite by assumption, so the full Cholesky decomposition can be obtained block by block as

$$\boldsymbol{C}=\begin{bmatrix}\boldsymbol{C}_1 & & \\ & \ddots & \\ & & \boldsymbol{C}_m\end{bmatrix}=\begin{bmatrix}\boldsymbol{L}_1 & & \\ & \ddots & \\ & & \boldsymbol{L}_m\end{bmatrix}\begin{bmatrix}\boldsymbol{L}_1^\top & & \\ & \ddots & \\ & & \boldsymbol{L}_m^\top\end{bmatrix}=\boldsymbol{L}\boldsymbol{L}^\top.$$

In this example, the product $\boldsymbol{D}^\top\boldsymbol{D}$ is a symmetric block tridiagonal matrix plus a multiple of the identity matrix. Despite the efficiency of extracting the spectral decomposition $\boldsymbol{L}^{-1}\boldsymbol{D}^\top\boldsymbol{D}(\boldsymbol{L}^\top)^{-1}=\boldsymbol{O}\Sigma\boldsymbol{O}^\top$ just once, this expensive operation remains a major computational bottleneck. In Table 1 results based on simultaneous diagonalization appear under the heading "Accelerated."

The fact that the Hessian $\boldsymbol{H}=\boldsymbol{C}+\rho\sum_{k=1}^3\alpha_k\boldsymbol{D}_k^\top\boldsymbol{D}_k$ is nearly block diagonal suggests the possibility of parallelism. To separate parameters by blocks, we set $\boldsymbol{p}_n=P_{S_2}[\boldsymbol{D}_2\operatorname{vec}(\boldsymbol{W}_n)]$ and $\bar{\boldsymbol{w}}_{nj}=\frac{1}{2}(\boldsymbol{w}_{n,j+1}+\boldsymbol{w}_{nj})$, exploit Jensen's inequality, and execute the majorization

$$\begin{aligned}\|\boldsymbol{D}_2\operatorname{vec}(\boldsymbol{W})-\boldsymbol{p}_n\|^2&=\sum_{j=1}^{m-1}\|\boldsymbol{w}_{j+1}-\boldsymbol{w}_j-\boldsymbol{p}_{nj}\|^2=\sum_{j=1}^{m-1}\left\|\frac{1}{2}(2\boldsymbol{w}_{j+1}-2\bar{\boldsymbol{w}}_{nj}-\boldsymbol{p}_n)-\frac{1}{2}(2\boldsymbol{w}_j-2\bar{\boldsymbol{w}}_{nj}+\boldsymbol{p}_{nj})\right\|^2\\&\le\frac{1}{2}\sum_{j=1}^{m-1}\|2\boldsymbol{w}_{j+1}-2\bar{\boldsymbol{w}}_{nj}-\boldsymbol{p}_n\|^2+\frac{1}{2}\sum_{j=1}^{m-1}\|2\boldsymbol{w}_j-2\bar{\boldsymbol{w}}_{nj}+\boldsymbol{p}_{nj}\|^2.\end{aligned}$$

Minimizing the block-separated surrogate amounts to solving in parallel m independent linear systems of the form

$$\left[\boldsymbol{C}_j+\gamma_j\boldsymbol{I}\right]\boldsymbol{w}_j=\boldsymbol{b}_j,\quad\gamma_j>0,$$

via the small spectral decompositions of the $\boldsymbol{C}_j$.

Table 1. Portfolio optimization with the S&P 500 data from 2004 to 2016 with the choices $\tau_1=\tau_2=1{,}000$ (30, 34)

Assets	Periods	Constraints		Iterations			Time (s)			Improvement	
s	m	ξ_{init}	ξ_{final}	Cholesky	Accelerated	Block	Cholesky	Accelerated	Block	Accelerated	Block
442	2	1	1.2100	1867	1867	2424	1.3	1.2	0.57	1.1	2.2
442	3	1	1.3310	2382	2382	2764	4.3	3.5	1.8	1.2	2.3
442	4	1	1.4641	3322	3321	3738	13	9.2	3.8	1.4	3.3
442	5	1	1.6105	4165	4165	5014	29	22	12	1.3	2.5
442	6	1	1.7716	4438	4438	5365	47	38	23	1.2	2.1
442	7	1	1.9487	4349	4349	4991	67	57	33	1.2	2.0
442	8	1	2.1436	5657	5653	7133	120	93	62	1.2	1.9

In each scenario, all three algorithms converge with a distance penalty $\le 10^{-6}$ and a feasibility residual $\|\boldsymbol{A}\operatorname{vec}(\boldsymbol{W})-\boldsymbol{b}\|\le 10^{-4}$. Times are reported with two significant digits.

Table 1 compares our accelerated and block-separated algorithms to the proximal distance algorithm based on the full Hessian's Cholesky decompositions. Because the latter must be recomputed every time ρ changes, both accelerated algorithms beat a naive proximal distance algorithm that solves a block diagonal linear system $\boldsymbol{H}\,\mathrm{vec}(\boldsymbol{W}) = \boldsymbol{b}$. The block-separated version leverages multiple cores on modern CPUs. In our example, we use two threads to carry out two block updates in parallel. Each thread in turn has four BLAS threads to carry out linear algebra operations, resulting in eight active cores during execution. Our parallel strategy leads to an appreciable twofold improvement in timing as the number of periods m increases, despite requiring more iterations.

3. Path Following

To illustrate the merits of path following, we revisit distance majorization under a fusion constraint $\boldsymbol{D}\boldsymbol{x} \in S$. For the surrogate [2], differentiation of the stationary condition

$$\boldsymbol{0} = \nabla f(\boldsymbol{x}) + \rho \boldsymbol{D}^\top [\boldsymbol{D}\boldsymbol{x} - P_S(\boldsymbol{D}\boldsymbol{x})],$$

with respect to ρ gives the path condition

$$\boldsymbol{0} = d^2 f(\boldsymbol{x}) d_\rho \boldsymbol{x} + \boldsymbol{D}^\top [\boldsymbol{D}\boldsymbol{x} - P_S(\boldsymbol{D}\boldsymbol{x})] + \rho \boldsymbol{D}^\top [\boldsymbol{D} d_\rho \boldsymbol{x} - dP_S(\boldsymbol{D}\boldsymbol{x}) \boldsymbol{D} d_\rho \boldsymbol{x}].$$

Solving for $d_\rho \boldsymbol{x} = \frac{d}{d\rho}\boldsymbol{x}$ yields

$$d_\rho \boldsymbol{x} = -\{d^2 f(\boldsymbol{x}) + \rho \boldsymbol{D}^\top [\boldsymbol{I} - dP_S(\boldsymbol{D}\boldsymbol{x})]\boldsymbol{D}\}^{-1} \boldsymbol{D}^\top [\boldsymbol{D}\boldsymbol{x} - P_S(\boldsymbol{D}\boldsymbol{x})].$$

The differential $dP_S(\boldsymbol{y})$ is known for projection onto an affine subspace. Indeed, let $\boldsymbol{A}$ be a matrix with full row rank and $S = \{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$ be an affine subspace. The closest point to $\boldsymbol{y}$ in S is

$$P_S(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{A}^\top)^{-1} (\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}).$$

The matrix $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}$ is an orthogonal projection. In this case, $dP_S(\boldsymbol{y}) = \boldsymbol{P}$. In general, if S is convex, then $\mathrm{dist}(\boldsymbol{y}, S)^2$ is a convex function, and Alexandrov's theorem (35) implies that $dP_S(\boldsymbol{y}) = -d^2 \frac{1}{2}\,\mathrm{dist}(\boldsymbol{y}, S)^2$ exists for almost all points $\boldsymbol{y}$. The differential $dP_S(\boldsymbol{y})$ also exists for almost all $\boldsymbol{y}$ (no coordinate ties) when S is a sparsity set.

For a general penalty function $p(\boldsymbol{x})$ with exponential penalty constants $\rho = e^\eta$, the stationary condition

$$\boldsymbol{0} = \nabla f(\boldsymbol{x}) + e^\eta \nabla p(\boldsymbol{x}).$$

gives after implicit differentiation and solving for η the equation

$$d_\eta \boldsymbol{x} = -[d^2 f(\boldsymbol{x}) + e^\eta d^2 p(\boldsymbol{x})]^{-1} e^\eta \nabla p(\boldsymbol{x}).$$

The update $\boldsymbol{x}_{\eta_{t+1}} = \boldsymbol{x}_{\eta_t} + d_\eta \boldsymbol{x}_{\eta_t}(\eta_{t+1} - \eta_t)$ is favored when $\eta_{n+1} = c\eta_n$ is incremented proportionally with $c > 1$.

Fig. 1 illustrates path following in a proximal distance algorithm for the fused lasso and portfolio optimization examples described in Examples 1.1 and 1.5, respectively. Extrapolating solution estimates at $\rho = \rho_n$ to $\rho = \rho_{n+1}$ keeps the estimates close to stationarity after perturbation, decreasing the total number of iterations needed to drive distance penalties downhill. The advantages of path following are more pronounced in Example 1.5 because it involves projection onto an affine set, which has a smooth differential. While we extrapolate solution estimates here, a reviewer astutely suggests that further improvement may be possible via extrapolation on the penalty parameter sequence. Similar ideas have succeeded in accelerating proximal gradient methods; see for instance section 4.3 of Parikh and Boyd (36).

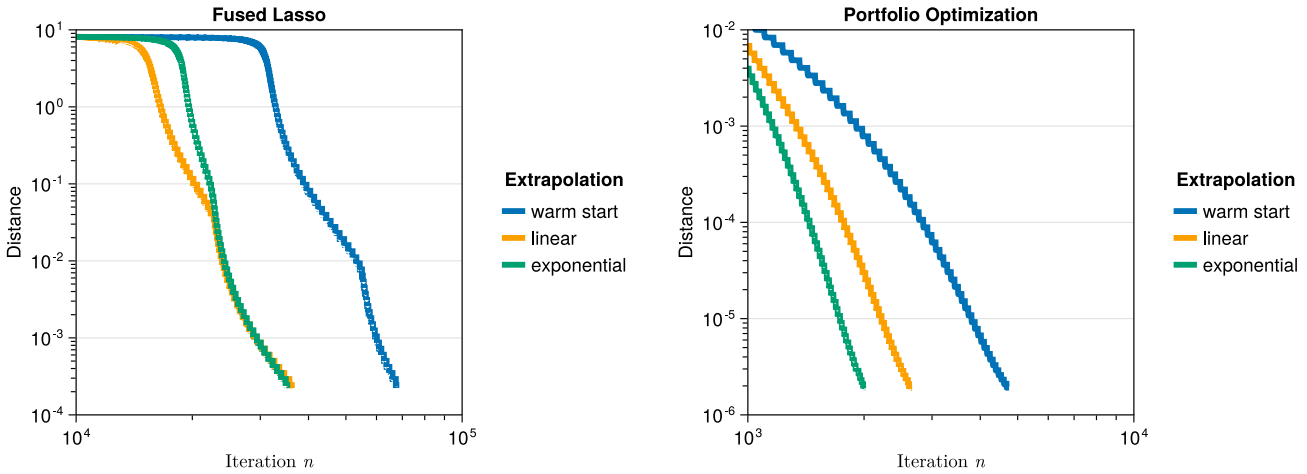

Fig. 1. Comparison of distance penalties under different extrapolation schemes for Example 1.1 (*Left*) and Example 1.5 (*Right*). Sharp changes indicate points where solutions are extrapolated from $\rho = \rho_n$ to $\rho = \rho_{n+1}$. Smooth changes indicate points where ρ is held constant.

4. Trust Regions

Trust regions are classical tools that act as safeguards in optimization when local quadratic approximations break down (37, 38). Cubic majorization achieves a similar goal even when the local approximate Hessian is indefinite. Here, we explore two generalizations of the MM principle based on local majorization.

A. Proximal Gradient Descent. Consider the task of minimizing $f(\boldsymbol{x})$ over the closed convex set C. Define the trust region $B_{2\|\boldsymbol{x}_n-\boldsymbol{x}_{n-1}\|}(\boldsymbol{x})$ where $\|\boldsymbol{x}-\boldsymbol{x}_n\| \leq 2\|\boldsymbol{x}_n-\boldsymbol{x}_{n-1}\|$, and suppose L is a Lipschitz constant for $\nabla f(\boldsymbol{x})$ on the bounded region. The projected gradient algorithm mandates the update

$$\boldsymbol{x}_{n+1} = P_C[\boldsymbol{x}_n - c^{-1}\nabla f(\boldsymbol{x}_n)],$$

for some $c \geq L$. Because projection onto a closed convex set is nonexpansive and because

$$\begin{aligned}\|\boldsymbol{x}_{n+1}-\boldsymbol{x}_n\| &= \|P_C[\boldsymbol{x}_n - c^{-1}\nabla f(\boldsymbol{x}_n)] - P_C[\boldsymbol{x}_{n-1} - c^{-1}\nabla f(\boldsymbol{x}_{n-1})]\| \leq \|\boldsymbol{x}_n - c^{-1}\nabla f(\boldsymbol{x}_n) - \boldsymbol{x}_{n-1} + c^{-1}\nabla f(\boldsymbol{x}_{n-1})\| \\ &\leq (1+c^{-1}L)\|\boldsymbol{x}_n-\boldsymbol{x}_{n-1}\|,\end{aligned}$$

the iterate $\boldsymbol{x}_{n+1}$ stays within the trust region $\|\boldsymbol{x}-\boldsymbol{x}_n\| \leq 2\|\boldsymbol{x}_n-\boldsymbol{x}_{n-1}\|$ where the Lipschitz constant is valid. This result makes it abundantly clear that $\nabla f(\boldsymbol{x})$ does not have to be globally Lipschitz to implement projected gradient descent. Taking $c = L$ is apt to promote the fastest convergence.

The same considerations apply to proximal gradient descent. In this case, one must minimize $g(\boldsymbol{x})+f(\boldsymbol{x})$ with $g(\boldsymbol{x})$ convex and $\nabla f(\boldsymbol{x})$ Lipschitz with constant L on the trust region $\|\boldsymbol{x}-\boldsymbol{x}_n\| \leq 2\|\boldsymbol{x}_n-\boldsymbol{x}_{n-1}\|$. Now the next iterate is $\boldsymbol{x}_{n+1} = \operatorname{prox}_g[\boldsymbol{x}_n - c^{-1}\nabla f(\boldsymbol{x}_n)]$. Because the proximal map is nonexpansive, the same argument shows that $\boldsymbol{x}_{n+1}$ falls within the trust region. Proximal distance algorithms with convex constraints conform to this paradigm because they can be viewed as proximal gradient descent algorithms (14).

B. Cubic Majorization. For a twice differentiable function $f(\boldsymbol{x})$ satisfying the Lipschitz inequality

$$\|d^2 f(\boldsymbol{v}) - d^2 f(\boldsymbol{u})\| \leq L\|\boldsymbol{v}-\boldsymbol{u}\|, \tag{10}$$

the Taylor expansion

$$f(\boldsymbol{x}+\boldsymbol{y}) = f(\boldsymbol{x}) + df(\boldsymbol{x})\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top d^2 f(\boldsymbol{x})\boldsymbol{y} + \int_0^1 \boldsymbol{y}^\top[d^2 f(\boldsymbol{x}+t\boldsymbol{y}) - d^2 f(\boldsymbol{x})]\boldsymbol{y}(1-t)\,dt,$$

leads to the majorization (39)

$$f(\boldsymbol{x}_n+\boldsymbol{y}) \leq f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top d^2 f(\boldsymbol{x}_n)\boldsymbol{y} + L\|\boldsymbol{y}\|^3\int_0^1 t(1-t)\,dt = f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top d^2 f(\boldsymbol{x}_n)\boldsymbol{y} + \frac{L}{6}\|\boldsymbol{y}\|^3, \tag{11}$$

upon invoking the Lipschitz condition [**10**]. Importantly, this cubic majorization is valid regardless of whether $d^2 f(\boldsymbol{x}_n)$ is positive definite. Cubic majorization is trivially suitable when $f(\boldsymbol{x})$ is a cubic polynomial, but it also applies to many other losses such as the negative log-likelihood in logistic regression (40).

At a stationary point $\boldsymbol{y} \in \mathbb{R}^p$ of the surrogate, we have

$$\boldsymbol{0} = \nabla f(\boldsymbol{x}_n) + d^2 f(\boldsymbol{x}_n)\boldsymbol{y} + \frac{L}{2}\|\boldsymbol{y}\|\boldsymbol{y},$$

or equivalently

$$\boldsymbol{y} = -\left[d^2 f(\boldsymbol{x}_n) + \frac{L}{2}\|\boldsymbol{y}\|\boldsymbol{I}\right]^{-1}\nabla f(\boldsymbol{x}_n). \tag{12}$$

Thus, the next iterate can be interpreted as a kind of damped Newton update. Taking the norm of this last equation yields the identity

$$\|\boldsymbol{y}\|^2 = df(\boldsymbol{x}_n)\left[d^2 f(\boldsymbol{x}_n) + \frac{L}{2}\boldsymbol{I}\|\boldsymbol{y}\|\right]^{-2}\nabla f(\boldsymbol{x}_n), \tag{13}$$

which is a univariate equation for the unknown $\|\boldsymbol{y}\|$. The easiest way of solving Eq. **13** is to replace $d^2 f(\boldsymbol{x}_n)$ by its spectral decomposition $\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top$. If we let $(u_i) = \boldsymbol{u} = \boldsymbol{V}^\top\nabla f(\boldsymbol{x}_n)$ and $\boldsymbol{D} = \operatorname{diag}(d_i)$, then Eq. **13** becomes

$$\|\boldsymbol{y}\|^2 = \sum_{i=1}^p \frac{u_i^2}{\left(d_i + \frac{L}{2}\|\boldsymbol{y}\|\right)^2}.$$

This equation possesses a unique positive root $\|\boldsymbol{y}\|$ (41) and provides the next iterate $\boldsymbol{x}_{n+1}$ via Eq. **12** and the identification $\boldsymbol{x}_{n+1} = \boldsymbol{y} + \boldsymbol{x}_n$.

A major barrier to cubic majorization is finding a local Lipschitz constant L. De Leeuw and Groenen (42) suggest MM algorithms for this problem. Secondary computational concerns include diagonalizing $d^2 f(\boldsymbol{x}_n)$ and finding the solution to Eq. **12**. These challenges have prompted research into adaptive cubic regularization (43). In any event, it is fruitful to view cubic majorization as a way of stabilizing the minimization of $f(\boldsymbol{x})$ when the second differential $d^2 f(\boldsymbol{x}_n)$ is indefinite, or the quadratic approximation is untrustworthy. In this sense, cubic majorization is akin to trust region methods (41).

Techniques from the method of trust regions can be exploited to avoid the diagonalization step. Suppose that we restrict minimization of the surrogate to the trust region $B_r(\boldsymbol{x}_n)$. On this ball, the cubic surrogate [**11**] itself is then majorized by the quadratic surrogate

$$h(\boldsymbol{x} \mid \boldsymbol{x}_n) = f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)(\boldsymbol{x} - \boldsymbol{x}_n) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_n)^\top d^2 f(\boldsymbol{x}_n)(\boldsymbol{x} - \boldsymbol{x}_n) + \frac{Lr_n}{6}\|\boldsymbol{x} - \boldsymbol{x}_n\|^2,$$

upon substituting r_n for one of the factors in the cubic $\frac{L}{6}\|\boldsymbol{x} - \boldsymbol{x}_n\|^3$. The quadratic surrogate $h(\boldsymbol{x} \mid \boldsymbol{x}_n)$ can be explicitly minimized to yield the update

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \left[d^2 f(\boldsymbol{x}_n) + \frac{Lr_n}{3}\boldsymbol{I}\right]^{-1} \nabla f(\boldsymbol{x}_n), \qquad [14]$$

assuming $d^2 f(\boldsymbol{x}_n) + \frac{Lr_n}{3}\boldsymbol{I}$ is positive definite. Once again, the MM principle ensures that minimizing the surrogate function $h(\boldsymbol{x} \mid \boldsymbol{x}_n)$ drives the quadratic approximation of $f(\boldsymbol{x})$ downhill. Taking $d_n = \frac{Lr_n}{3}$, multiplying by $\left[d^2 f(\boldsymbol{x}_n) + d_n \boldsymbol{I}\right]$, and rearranging the update formula reveals that the update [**14**] approximates the implicit gradient descent step

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{d_n}\left[df(\boldsymbol{x}_n) + d^2 f(\boldsymbol{x}_n)(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)\right] \quad \approx \quad \boldsymbol{x}_n - \frac{1}{d_n}\nabla f(\boldsymbol{x}_{n+1}).$$

The update [**14**] also has the advantage of substituting a linear solve for a diagonalization. Unfortunately, it also faces two potential hurdles. First, contrary to our assumption, the Hessian $d^2 f(\boldsymbol{x}_n) + \frac{Lr_n}{3}\boldsymbol{I}$ may be indefinite. Second, the proposed update $\boldsymbol{x}_{n+1}$ may fall outside the trust region.

These issues can be remedied by building upon a recent idea of Mishchenko (44). We now show that the *Right* choice of the radius r_n forces $\boldsymbol{x}_{n+1}$ to fall within the trust region. If $d^2 f(\boldsymbol{x}_n)$ fails to be positive semidefinite, then we replace it by $\boldsymbol{A}_n = d^2 f(\boldsymbol{x}_n) - \lambda \boldsymbol{I}$, where λ falls below the smallest eigenvalue of $d^2 f(\boldsymbol{x}_n)$. In some cases, a known lower bound on λ can be exploited. For instance, when $f(\boldsymbol{x})$ is convex, the smallest eigenvalue is bounded below by 0. To compensate for subtracting $\lambda \boldsymbol{I}$ from $d^2 f(\boldsymbol{x})$, we add back $\lambda \boldsymbol{I}$ in the cubic majorization term.

Mishchenko makes the inspired choice $r_n = c_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}$ for some positive constant c_n to be specified. To implement the idea, we will choose a constant $d_n > 0$ satisfying

$$\lambda + \frac{Lr_n}{3} \le d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}.$$

This choice supplies the cubic majorization

$$f(\boldsymbol{x}_n + \boldsymbol{y}) \le f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{A}_n \boldsymbol{y} + \frac{1}{2}\left(\lambda + \frac{Lr_n}{3}\right)\|\boldsymbol{y}\|^2 \le f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{A}_n \boldsymbol{y} + \frac{d_n}{2}\sqrt{\|\nabla f(\boldsymbol{x}_n)\|} \cdot \|\boldsymbol{y}\|^2.$$

The analogue of the update [**14**] now amounts to

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \left[\boldsymbol{A}_n + d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}\boldsymbol{I}\right]^{-1} \nabla f(\boldsymbol{x}_n), \qquad [15]$$

where the inverse is well-defined because $\boldsymbol{A}_n$ is positive definite. Near the minimum point, $d^2 f(\boldsymbol{x}_n)$ will ordinarily be positive definite and $\lambda = 0$. Furthermore, $\frac{Lc_n}{3} \ge \frac{Lc_n}{3} + \lambda$, so in practice we simply require $d_n \ge \frac{Lc_n}{3}$.

Fortunately, Mishchenko's foresight ensures that this update stays within the trust region. To understand why, we observe that the matrix $(\boldsymbol{A}_n + d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}\boldsymbol{I})^{-1}$ is less positive definite than the matrix $d_n^{-1}\|\nabla f(\boldsymbol{x}_n)\|^{-1/2}\boldsymbol{I}$. This fact is reflected in the sizes of their spectral norms and yields the following bound

$$\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| = \left\|\left(\boldsymbol{A}_n + d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}\boldsymbol{I}\right)^{-1}\nabla f(\boldsymbol{x}_n)\right\| \le \frac{1}{d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}}\|\nabla f(\boldsymbol{x}_n)\| = \frac{1}{d_n}\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}.$$

Here $\frac{1}{d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}}$ is the spectral norm of the matrix $d_n^{-1}\|\nabla f(\boldsymbol{x}_n)\|^{-1/2}\boldsymbol{I}$. The bound is less than the trust region radius r_n provided $d_n^{-1} \le c_n$. To mimic Newton's method, we set $d_n = \min_{c_n} \max\{\frac{Lc_n}{3}, c_n^{-1}\}$. The minimum value $\sqrt{L/3}$ is achieved when $c_n = \sqrt{3/L}$. This translates into the trust region radius

$$r_n = \sqrt{\frac{3}{L}\|\nabla f(\boldsymbol{x}_n)\|}.$$

We emphasize that the Lipschitz constant L need not be global; its validity on the trust region suffices for this argument. In the general setting where no global Lipschitz constant exists, there is a circular dependency between c_n, d_n, and the local Lipschitz constant L_{r_n}. In practice, a coarse estimate of L_{r_n} based on a generous overestimate of the radius r_n usually suffices and avoids solving a highly nonlinear equation. In any event, the overall algorithm achieves a superlinear rate of convergence because $d_n\sqrt{\|\nabla f(\boldsymbol{x}_n)\|}$ tends to 0 as $\boldsymbol{x}_n$ approaches the minimum point.

To overcome the reader's skepticism about the ease of finding bounds and Lipschitz constants, consider the case of a monomial $m(\boldsymbol{x}) = \prod_j x_j^{d_j}$ of degree $d = \sum_j d_j \geq 1$ over the ball $B_r(\mathbf{0})$ of radius r around the origin. Following de Leeuw and Groenen (42), we first attack the problem of bounding $m(\boldsymbol{x})$. It suffices to assume all coordinates are positive and maximize $\ln m(\boldsymbol{x}) = \frac{1}{2}\sum_j d_j \ln(x_j^2)$ subject to $\sum_j x_j^2 = r^2$. The maximum is attained when $x_j^2 = \frac{r^2 d_j}{d}$, giving an overall upper bound $|m(\boldsymbol{x})| \leq (\frac{r}{\sqrt{d}})^d \prod_j d_j^{d_j/2}$. As for the Lipschitz constant, we claim that $|m(\boldsymbol{x}) - m(\boldsymbol{y})| \leq dr^{d-1}\|\boldsymbol{x} - \boldsymbol{y}\|$. To prove this claim, note that it is obvious when $d = 1$ and that otherwise $m(\boldsymbol{x}) = x_j p(\boldsymbol{x})$ for some component x_j and monomial $p(\boldsymbol{x})$ of degree $d - 1$. It follows by induction on d that

$$|m(\boldsymbol{x}) - m(\boldsymbol{y})| = |(x_j - y_j)p(\boldsymbol{x}) + y_j[p(\boldsymbol{x}) - p(\boldsymbol{y})]| \quad \leq \quad dr^{d-1}\|\boldsymbol{x} - \boldsymbol{y}\|,$$

on the ball. These results apply to both forms of local majorization because the entries of $\nabla f(\boldsymbol{x})$ and $d^2 f(\boldsymbol{x})$ are also polynomials whenever $f(\boldsymbol{x})$ is a polynomial.

Let us illustrate the trust region Newton method outlined here with the Styblinski–Tang function described in Example 1.6. Since the Hessian is diagonal with diagonal entries $6x_i^2 - 16$, on $B_r(\mathbf{0})$ the local Lipschitz bound

$$\|d^2 f(\boldsymbol{x}) - d^2 f(\boldsymbol{y})\| = \|\operatorname{diag}(6x_i^2 - 16) - \operatorname{diag}(6y_i^2 - 16)\| = 6\|\operatorname{diag}[x_i^2 - y_i^2]\| \leq 12r\,\|\boldsymbol{x} - \boldsymbol{y}\|,$$

is available. Our proposed trust region method succeeds whenever $\boldsymbol{x}_0$ is initialized in the hypercube $[-5, 0]^d$, whereas the standard step-halving Newton's method may be trapped by an inferior local minimum. The trust region may fail to converge to the global minimum when initialized in the larger hypercube $[-5, 5]^d$. Table 2 contrasts the two methods starting from $\boldsymbol{x} = (-2.825, -0.156, -0.47, -3.205)$. Although both methods ultimately converge to the global minimum, cubic regularization avoids sharp increases in objective values.

Revisiting Example 1.7, an approximate trust region method can be used to minimize the negative log-likelihood in Poisson regression and thus recover the maximum likelihood estimate. For any two points $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in the trust region $B_r(\boldsymbol{\beta}_n)$, the spectral norm definition, the mean value inequality, and the Cauchy–Schwarz inequality together imply

$$\begin{aligned}
\|J(\boldsymbol{\beta}) - J(\boldsymbol{\gamma})\| = \|\boldsymbol{X}^\top[\boldsymbol{W}(\boldsymbol{\beta}) - \boldsymbol{W}(\boldsymbol{\gamma})]\boldsymbol{X}\| \quad &\leq \quad \|\boldsymbol{X}\|^2 \cdot \|\boldsymbol{W}(\boldsymbol{\beta}) - \boldsymbol{W}(\boldsymbol{\gamma})\| \\
|W_{ii}(\boldsymbol{\beta}) - W_{ii}(\boldsymbol{\gamma})| &\leq \max_s \left|\frac{d}{ds}\frac{\mu'(s)^2}{\mu(s)}\right| \cdot \|\boldsymbol{x}_i\| \cdot \|\boldsymbol{\beta} - \boldsymbol{\gamma}\| \\
\|\boldsymbol{W}(\boldsymbol{\beta}) - \boldsymbol{W}(\boldsymbol{\gamma})\| &\leq \max_s \left|\frac{d}{ds}\frac{\mu'(s)^2}{\mu(s)}\right| \cdot \max_i \|\boldsymbol{x}_i\| \cdot \|\boldsymbol{\beta} - \boldsymbol{\gamma}\|.
\end{aligned}$$

Under the canonical inverse link $\mu(s) = e^s$, the value $\max_s \left|\frac{d}{ds}\frac{\mu'(s)^2}{\mu(s)}\right|$ is unfortunately ∞. However, there is nothing sacred about the canonical link. The alternative inverse link

$$\mu(s) = \begin{cases} e^s & s \leq p \\ e^p\left[1 + (s-p) + \frac{(s-p)^2}{2} + \frac{(s-p)^3}{6}\right] & s > p \end{cases},$$

Table 2. Minimization iterates for the Styblinski–Tang function with $d = 4$

Iterate	Cubic regularization		Newton	
n	$f(\boldsymbol{x}_n)$	$\nabla f(\boldsymbol{x}_n)$	$f(\boldsymbol{x}_n)$	$\nabla f(\boldsymbol{x}_n)$
0	−80.0	16.52	−80.0	16.52
1	−130.43	24.26	7.17×10^{16}	1.47×10^{13}
2	−146.71	32.12	1.38×10^{18}	1.31×10^{14}
3	−155.37	10.28	2.72×10^{17}	3.87×10^{13}
4	−156.60	2.226	5.37×10^{16}	1.15×10^{13}
5	−156.66	0.2084	1.06×10^{16}	3.40×10^{12}
6	−156.66	3.894×10^{-3}	2.09×10^{15}	1.01×10^{12}
7	−156.66	4.638×10^{-6}	4.14×10^{14}	2.98×10^{11}
8	−156.66	5.952×10^{-11}	8.17×10^{13}	8.84×10^{10}
9	−156.66	1.005×10^{-14}	1.61×10^{13}	2.62×10^{10}
10	−156.66	1.005×10^{-14}	3.188×10^{12}	7.76×10^{9}
20	−156.66	1.005×10^{-14}	2.847×10^{5}	4.037×10^{4}
30	−156.66	1.005×10^{-14}	−156.66	1.05×10^{-10}

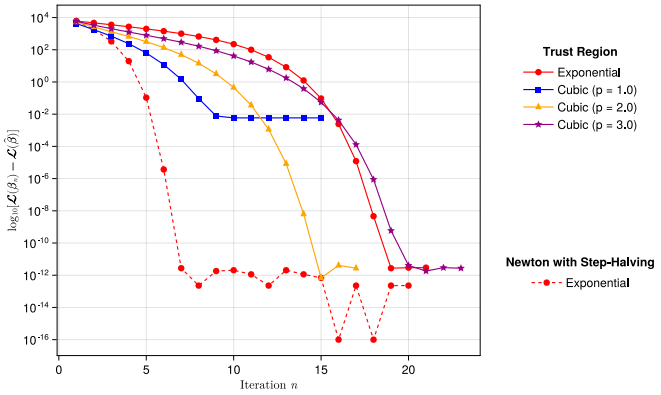

Fig. 2. Poisson regression on m = 1,000 samples and 10 predictors. For each inverse link function $\mu(s)$, the deviation $|\mathcal{L}(\boldsymbol{\beta}_n) - \mathcal{L}(\hat{\boldsymbol{\beta}})|$ from the corresponding maximum log-likelihood $\mathcal{L}(\hat{\boldsymbol{\beta}})$ is plotted on a semilogarithmic scale. All methods converge, but with slightly different log-likelihoods. Each predictor vector $\boldsymbol{x}_i$ was drawn from $\mathcal{N}(\mathbf{0}, \Sigma)$ under the weak but constant correlation model of Hardin et al. (45). This produces a design matrix $\boldsymbol{X}$ with condition number close to 1. The choice $\beta_i \sim \text{Uniform}(0, 0.05)$ generates responses y_i in $[0, 10]$ with mean and variance ≈ 3 under the exponential inverse link.

smoothly bridges the change point $p \geq 0$ up to its third derivative and yields empirically the global Lipschitz constant $1.623\, e^{\,p}\, \|\boldsymbol{X}\|^2 \max_i \|\boldsymbol{x}_i\|$ for the expected information. The trust region method has the virtue of almost always ascending and thus avoiding step halving. The change point $p = 1$ in the cubic model gives a better Lipschitz constant and keeps convergence of the trust region method on a par with Newton's method; Fig. 2. In contrast, the change point $p = 3$ decelerates convergence but improves the cubic approximation and produces a final log-likelihood comparable to that under the canonical link. The choice $p = 2$ for the alternative inverse link offers a good compromise between convergence speed and accuracy. Interestingly, parameter estimates under each approximation are comparable to the model under the canonical link, achieving a mean squared error on the order of 10^{-3}.

5. Discussion

Over time, we expect even wider adoption of the MM principle in statistics, machine learning, imaging, and other data-centric disciplines. We are still at the stage where the majority of data scientists are more aware of EM than the MM principle. This relative ignorance, which limits application to missing data settings, will abate as data scientists face the growing challenges of parameter estimation and model selection in high-dimensional problems. The simplicity and versatility of MM makes it an invaluable wedge in tackling these challenges. As we have witnessed, the proximal distance algorithm enables constrained estimation and fusion constraints $\boldsymbol{D}\boldsymbol{x} \in C$, where $\boldsymbol{D}$ is a matrix and C is a closed set. Computational algorithms work behind the scenes as part of the infrastructure of statistics and data science. They enable statistical analysis and constitute the lenses through which data is viewed. This is not the place to list exhaustively the applications of the MM principle. Suffice to say that the list includes diverse fields such as a) computed X-ray and positron tomography (46, 47), b) multidimensional scaling (48), c) construction of gene and protein networks (49), d) ranking of sports teams by the Bradley–Terry model (50), e) item response models such as the Rasch model (51), and f) matrix completion problems (52).

Our emphasis on efficient updating schemes and path following highlights the need for acceleration of proximal distance algorithms. Because these algorithms can be understood as annealing their way toward a constrained optimum, they can often take hundreds of iterations. The de facto computational costs of proximal distance algorithms represent a tradeoff between the cost per iteration and the number of iterations. Efficient updating lowers the cost per iteration, while path following lowers the number of iterations. Fortunately, as mentioned in the introduction, generic accelerators also apply (6–8, 10). Recent research also suggests the potential of stochastic approximation methods in scaling optimization methods via subsampling (53). On the other hand, the slow pace of annealing can also be viewed as a virtue, as it allows an algorithm to snake its way to a good solution, smoothing out and avoiding poor local minima in nonconvex problems. One criticism of Courant's penalty method is that it is susceptible to ill conditioning as the penalty constant ρ grows. Loss functions that reduce to least squares criteria effectively reduce the condition number from $O(\rho)$ to $O(\sqrt{\rho})$. The proximal distance surrogate promotes this reduction by augmenting the least squares loss.

The trust region methods we sketch lead to MM algorithms that can leverage local rather than global majorization. This generalization allows quadratic surrogates for functions that grow rapidly as $\|\boldsymbol{x}\| \to \infty$. Quadratic surrogates are ideal in allowing users to harness many useful tools from linear algebra. The crux of the matter in implementing these advances lies in finding appropriate Lipschitz constants. Fortunately, these do not have to be sharp to elicit effective algorithms. By taking a closer look at the various rules governing Lipschitz functions, progress toward automatic computation of local Lipschitz constants should be possible.

Finally, we are acutely aware of the need for further theory to bolster the experimental promise of proximal distance algorithms. Understanding algorithm behavior is apt to be very challenging once we depart from the comfortable confines of convexity, though there are grounds for optimism based on recent advances in real algebraic geometry applied to nonconvex optimization (54). The idea of distance penalization is also worth exploring further in adjacent areas. For instance, recent research highlights the utility of distance penalties in neural net modeling (55). Furthermore, distance penalties may eventually play a more prominent role in Bayesian inference in the form of log priors enforcing constraint satisfaction (56–58). These recent forays into Bayesian computation are incomplete, and there remain many interesting open questions related to inference with distance penalized priors. In the meantime, the MM perspective provides computable and stable (59) solutions to many challenging problems in data science. Its emphasis on core

computational tools and geometric insight will continue to illuminate many canonical problems of statistics and machine learning. We encourage readers to join us in advancing the theory and applications of the MM and proximal distance principles.

Data, Materials, and Software Availability. Open-source Julia code for our various examples is available at https://github.com/alanderos91/MMOptimizationAlgorithms (60). The Github repository is publicly available and documents simulation codes, scripts, and additional instructions to access third-party data.

1. K. Lange, *Numerical Analysis for Statisticians* (Springer, New York, NY, ed. 2, 2010).
2. K. Lange, *Optimization* (Springer, ed. 2, 2012).
3. J. Mairal, Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM J. Opt.* **25**, 829–855 (2015).
4. Y. Sun, P. Babu, D. P. Palomar, Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Trans. Sig. Process.* **65**, 794–816 (2017).
5. K. Lange, *MM Optimization Algorithms* (SIAM, 2016).
6. M. Agarwal, J. Xu, Quasi-Newton acceleration of EM and MM algorithms via Broyden's method. arXiv [Preprint] (2022). http://arxiv.org/abs/2201.05935 (Accessed 10 January 2023).
7. Y. Du, R. Varadhan, SQUAREM: An R Package for off-the-shelf acceleration of EM, MM and other EM-like monotone algorithms. *J. Stat. Software* **92**, 1–41 (2020).
8. N. C. Henderson, R. Varadhan, Damped Anderson acceleration with restarts and monotonicity control for accelerating EM and EM-like algorithms. *J. Comput. Graphical Stat.* **28**, 834–46 (2019).
9. R. Varadhan, C. Roland, Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scand. J. Stat.* **35**, 335–53 (2008).
10. H. Zhou, D. Alexander, K. Lange, A quasi-Newton acceleration for high-dimensional optimization algorithms. *Stat. Comput.* **21**, 261–273 (2011).
11. A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Stat. Soc. B* **39**, 1–38 (1977).
12. I. Olkin, A. W. Marshall, B. Arnold, *Inequalities: Theory of Majorization and Its Applications* (Academic Press, 2011).
13. E. C. Chi, H. Zhou, K. Lange, Distance majorization and its applications. *Math. Prog. A* **146**, 409–436 (2014).
14. K. L. Keys, H. Zhou, K. Lange, Proximal distance algorithms: Theory and practice. *J. Mach. Learn. Res.* **20**, 2384–2421 (2020).
15. J. Xu, E. C. Chi, K. Lange, Generalized linear model regression under distance-to-set penalties. *Adv. Neural Inf. Process. Syst.* **30**, 1385–1395 (2017).
16. J. Xu, E. C. Chi, M. Yang, K. Lange, An MM algorithm for split feasibility problems. *Comput. Optim. Appl.* **71**, 795–828 (2018).
17. J. Xu, K. Lange, A proximal distance algorithm for likelihood-based sparse covariance estimation. *Biometrika* **109**, 1047–1066 (2022).
18. H. Zhou, L. Hu, J. Zhou, K. Lange, MM algorithms for variance components models. *J. Comput. Grap. Stat.* **28**, 350–361 (2019).
19. R. Courant, *Variational Methods for the Solution of Problems of Equilibrium and Vibrations* (Verlag Nicht Ermittelbar, 1943).
20. E. J. Beltrami, An Algorithmic Approach to Nonlinear Analysis and Optimization (Academic Press, 1970).
21. H. H. Bauschke, P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (Springer, 2011).
22. A. Beck, *First-Order Methods in Optimization* (SIAM, 2017).
23. A. Landeros, O. H. M. Padilla, H. Zhou, K. Lange, Extensions to the proximal distance of method of constrained optimization. *J. Mach. Learn. Res.* **23**, 1–45 (2022).
24. D. Böhning, B. G. Lindsay, Monotonicity of quadratic approximation algorithms. *Ann. Inst. Stat. Math.* **40**, 641–663 (1988).
25. R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. B* **67**, 91–108 (2005).
26. J. P. Boyle, R. L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert spaces. *Lect. Notes Stat.* **37**, 28–47 (1986).
27. J. H. Won, J. Xu, K. Lange, "Projection onto Minkowski sums with application to constrained learning" in *International Conference on Machine Learning* (2019), pp. 3642–3651.
28. J. Bien, R. J. Tibshirani, Sparse estimation of a covariance matrix. *Biometrika* **98**, 807–820 (2011).
29. R. H. Bartels, G. W. Stewart, Solution of the matrix equation $AX + XB = C$. *Commun. ACM* **15**, 820–826 (1972).
30. V. De Simone, D. di Serafino, J. Gondzio, S. Pougkakiotis, M. Viola, Sparse approximations with interior point methods. *SIAM Rev.* **64**, 954–88 (2022).
31. M. A. Styblinski, T. S. Tang, Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing. *Neural Networks* **3**, 467–483 (1990).
32. M. Jamil, X. S. Yang, A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Modell. Numer. Optim.* **4**, 150–194 (2013).
33. M. Seeger, "Low rank updates for the Cholesky decomposition" (Tech. Rep., University of California, Berkeley, CA, 2004).
34. R. Bruni, F. Cesarone, A. Scozzani, F. Tardella, Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models. *Data Brief* **8**, 858–862 (2016).
35. C. P. Niculescu, L.-E. Persson, *Convex Functions and Their Applications: A Contemporary Approach* (Springer, 2005).
36. N. Parikh, S. Boyd, Proximal algorithms. *Found. Trends Opt.* **1**, 127–239 (2014).
37. J. E. Dennis Jr., R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM, 1966).
38. J. Nocedal, S. J. Wright, *Numerical Optimization* (Springer, ed. 2, 2006).
39. Y. Nesterov, B. Polyak, "Cubic regularization of a Newton scheme and its global performance" (CORE Discussion Paper 41, Catholic University of Louvain, 2003).
40. J. De Leeuw, "Quadratic and cubic majorization" (Preprint Series, UCLA Department of Statistics Department of Statistics, 2006).
41. A. R. Conn, N. I. M. Gould, P. L. Toint, *Trust Region Methods* (SIAM, 2000).
42. J. De Leeuw, P. J. F. Groenen, "Majorizing a multivariate polynomial over the unit sphere, with applications" (Preprint Series, UCLA Department of Statistics, 2011).
43. C. Cartis, N. I. M. Gould, P. L. Toint, An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function–evaluation complexity. *IMA J. Numer. Anal.* **32**, 1662–1695 (2012).
44. K. Mishchenko, Regularized Newton method with global $O(1/k^2)$ convergence. arXiv [Preprint] (2021), http://arxiv.org/abs/2112.02089 (Accessed 1 October 2022).
45. J. Hardin, S. R. Garcia, D. Golan, A method for generating realistic correlation matrices. *Ann. Appl. Stat.* **7**, 1733–1762 (2013).
46. K. Lange, M. Bahn, R. Little, A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *IEEE Trans. Med. Imaging* **6**, 106–114 (1987).
47. K. Lange, R. Carson, EM reconstruction algorithms for emission and transmission tomography. *J. Comput. Assist. Tomogr.* **8**, 306–316 (1984).
48. P. J. F. Groenen, R. Mathar, W. J. Heiser, The majorization approach to multidimensional scaling for Minkowski distances. *J. Class.* **12**, 3–19 (1995).
49. J. M. Ranola, S. Ahn, M. E. Sehl, D. J. Smith, K. Lange, A Poisson model for random multigraphs. *Bioinformatics* **26**, 2004–2011 (2010).
50. D. R. Hunter, MM algorithms for generalized Bradley–Terry models. *Ann. Stat.* **32**, 384–406 (2004).
51. J. De Leeuw, Verhelst maximum likelihood estimation in generalized Rasch models. *J. Ed. Stat.* **3**, 183–196 (1986).
52. T. Hastie, R. Mazumder, J. Lee, R. Zadeh, Matrix completion and low-rank SVD via fast alternating least squares. *J. Mach. Learn. Res.* **16**, 3367–3402 (2014).
53. H. Jiang, J. Xu, The stochastic proximal distance algorithm. arXiv [Preprint] (2022). http://arxiv.org/abs/2210.12277 (Accessed 10 January 2023).
54. J. B. Lasserre, *An Introduction to Polynomial and Semi-Algebraic Optimization* (Cambridge University Press, 2015).
55. B. Peters, Point-to-set distance functions for output-constrained neural networks. *J. Appl. Numer. Opt.* **4**, 175–201 (2022).
56. Q. Heng, H. Zhou, E. C. Chi, Bayesian trend filtering via proximal Markov chain Monte Carlo. arXiv [Preprint] (2022). http://arxiv.org/abs/2201.00092 (Accessed 10 January 2023).
57. R. Presman, J. Xu, Distance-to-set priors and constrained Bayesian inference. arXiv [Preprint] (2022). http://arxiv.org/abs/2210.12258 (Accessed 10 January 2023).
58. M. Xu, H. Zhou, Y. Hu, L. Duan, Bayesian inference using the proximal mapping: Uncertainty quantification under varying dimensionality. arXiv [Preprint] (2021). http://arxiv.org/abs/2108.04851 (Accessed 10 January 2023).
59. B. Yu, K. Kumbier, Veridical data science. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 3920–3929 (2020).
60. A. Landeros, J. Xu, K. Lange, MM optimization examples. Github. https://github.com/alanderos91/MMOptimizationAlgorithms. Deposited 18 April 2023.