Mathematical Programming manuscript No.
(will be inserted by the editor)

Level-set methods for convex optimization

Aleksandr Y. Aravkin · James V. Burke · Dmitry Drusvyatskiy · Michael P. Friedlander · Scott Roy

Received: date / Accepted: date

Abstract Convex optimization problems arising in applications often have favorable objective functions and complicated constraints, thereby precluding first-order methods from being immediately applicable. We describe an approach that exchanges the roles of the objective with one of the constraint functions, and instead approximately solves a sequence of parametric level-set problems. Two Newton-like zero-finding procedures for nonsmooth convex functions, based on inexact evaluations and sensitivity information, are introduced. It is shown that they lead to efficient solution schemes for the original problem. We describe the theoretical and practical properties of this approach for a broad range of problems, including low-rank semidefinite optimization, sparse optimization, and gauge optimization.

Keywords first-order methods · root-finding · semidefinite & conic programming

A.Y. Aravkin
Applied Mathematics, University of Washington, Seattle WA
E-mail: saravkin@uw.edu
Research supported by the Washington Research Foundation Data Science Professorship.

J.V. Burke
E-mail: jvburke01@gmail.com
Research supported in part by the NSF award DMS-1514559.

D. Drusvyatskiy
Mathematics, University of Washington, Seattle WA
E-mail: ddrusv@uw.edu
Research supported by the AFOSR YIP award FA9550-15-1-0237.

M.P. Friedlander
Computer Science, University of British Columbia, Vancouver BC
E-mail: mpf@ubc.ca
Research supported by the ONR award N00014-16-1-2242.

S. Roy
Mathematics, University of Washington, Seattle WA
E-mail: scott.michael.roy@gmail.com
Research supported in part by the AFOSR YIP award FA9550-15-1-0237.

1 Introduction

We demonstrate a method for solving constrained convex optimization problems that interchanges the objective with one of the constraint functions. This interchange defines a convex and nonsmooth univariate optimal-value function $v(\tau)$, which is parameterized by the level values of the original objective. A solution of the original problem can then be obtained by computing a root τ_* of a single nonlinear equation of the form

$$v(\tau) = \sigma, \tag{1.1}$$

where the root corresponds to the desired optimal level value. This approach has been used to develop a variety of solution methods—some dating back to antiquity; see §1.3. Particular implementations of this idea, however, are all tied to specific choices of the algorithm used to approximate the function value $v(\tau_k)$ and the algorithm used to update the sequence of level values $\tau_k \to \tau_*$. Our proposed approach only requires a fixed relative accuracy between upper and lower bounds on $v(\tau_k)$. In doing so, we give an algorithm with an overall iteration complexity that is only a log factor of the iteration complexity needed to approximate $v(\tau_k)$. This results in a framework that decouples the method for approximating $v(\tau)$ from the method for solving (1.1) and thereby allows for the specification of a wide range of new approaches to constrained convex optimization.

The story behind our approach begins with the SPGL1 algorithm for basis pursuit [52, 53]. Although neither SPGL1 nor basis pursuit are our focus, they provide a concrete illustration of the ideas we pursue. Recall that the goal of the basis pursuit problem is to recover a sparse n-vector x that approximately satisfies the linear system $Ax = b$. This task often arises in applications such as compressed sensing and statistical model selection. Standard approaches, based on convex optimization, rely on solving one of the following formulations.

BP_σ	LS_τ	QP_λ
$\min_x \ \|x\|_1$ s.t. $\tfrac{1}{2}\|Ax-b\|_2^2 \le \sigma$	$\min_x \ \tfrac{1}{2}\|Ax-b\|_2^2$ s.t. $\|x\|_1 \le \tau$	$\min_x \ \tfrac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1$

Computationally, BP_σ is perceived to be the most challenging of the three formulations because of the complicated geometry of the feasible region. For example, projected- or proximal-gradient methods for LS_τ or QP_λ require at each iteration applications of the operator A and its adjoint, and computing either a Euclidean projection onto the 1-norm ball or a proximal step, which cost $\mathcal{O}(n \log n)$ and $\mathcal{O}(n)$ operations, respectively. As a result, solvers such as FISTA [3] and SPARSA [58], target either LS_τ and QP_λ. The Homotopy algorithm [44] and alternating direction method of multipliers (ADMM) [12, 27] can be applied in various ways to solve BP_σ, but available implementations require solving a linear system at each iteration, which is not always practical for large problems. Inexact variants of ADMM, such as linearized Bregman [59], do not require a linear solution, but may compromise by solving an approximation of the problem [59].

This paper targets optimization problems that generalize the formulations BP_σ and LS_τ. To set the stage, consider the pair of convex problems

$$\begin{aligned} &\underset{x\in\mathcal{X}}{\text{minimize}} \quad \varphi(x) &&\text{subject to} \quad \rho(Ax-b)\le\sigma, &&(\mathcal{P}_\sigma)\\ &\underset{x\in\mathcal{X}}{\text{minimize}} \quad \rho(Ax-b) &&\text{subject to} \quad \varphi(x)\le\tau, &&(\mathcal{Q}_\tau)\end{aligned}$$

where $\mathcal{X}\subseteq\mathbb{R}^n$ is a closed convex set, the functions $\varphi:\mathbb{R}^n\to\overline{\mathbb{R}}:=\mathbb{R}\cup\{+\infty\}$ and $\rho:\mathbb{R}^n\to\overline{\mathbb{R}}$ are closed convex functions, and A is a linear map. Such formulations are ubiquitous in contemporary optimization and its applications, and often φ may be regarded as a regularizer on the solution x, and ρ may be regarded as a measure of misfit between a linear model Ax and observations b. Other formulations are available, of course, that may be more natural in a particular context (such as redefining ρ so that A and b are not explicit). We choose this formulation because it most closely represents a large class of problems that might appear in practice.

Our working assumption is that the level-set problem $\mathcal{Q}_\tau$ is easier to solve than $\mathcal{P}_\sigma$ in the sense that there exists a specialized algorithm for its solution, but that a comparably-efficient solver does not exist for $\mathcal{Q}_\tau$. In §4, we discuss a range of optimization problems, including problems with nonsmooth regularization, and conic constraints, that have this property.

Our main contribution is to develop a practical and theoretically rigorous algorithmic framework to harness existing algorithms for $\mathcal{Q}_\tau$ to efficiently solve the $\mathcal{P}_\sigma$ formulation. As a consequence, we make explicit the fact that in typical circumstances both problems are essentially equivalent from the viewpoint of computational complexity. Hence, there is no reason not to choose any one preferred formulation based on computational considerations alone. This observation is very significant in applications since, although the formulations $\mathcal{P}_\sigma$, $\mathcal{Q}_\tau$, and their penalty-function counterparts are, in a sense, mathematically and computationally equivalent, they are far from equivalent from a modeling perspective. Practitioners should instead focus on choosing the formulation best suited to their applications. Our second contribution is to provide an algorithmic recipe for achieving the same computational complexity for a wide range of regularized data-fitting problems, listed in §1.2.

1.1 Approach

The proposed approach, which we will formalize shortly, approximately solves $\mathcal{P}_\sigma$ in the sense that it generates a point $x\in\mathcal{X}$ that is *super-optimal* and *ϵ-feasible*:

$$\varphi(x)\le\mathrm{OPT} \qquad\text{and}\qquad \rho(Ax-b)\le\sigma+\epsilon, \tag{1.2}$$

where OPT is the optimal value of $\mathcal{P}_\sigma$. This optimality concept was introduced by Harchaoui et al. [28], and we adopt it here. Our proposed strategy exchanges the roles of the objective and constraint functions in $\mathcal{P}_\sigma$, and approximately solves a sequence of level-set problems $\mathcal{Q}_\tau$ for varying parameters τ. There are many precursors for this level set approach, which we summarize in §1.3.

How does one use approximate solutions of $\mathcal{Q}_\tau$ to obtain a super-optimal and ϵ-feasible solution of $\mathcal{P}_\sigma$, the target problem? We answer this by recasting the

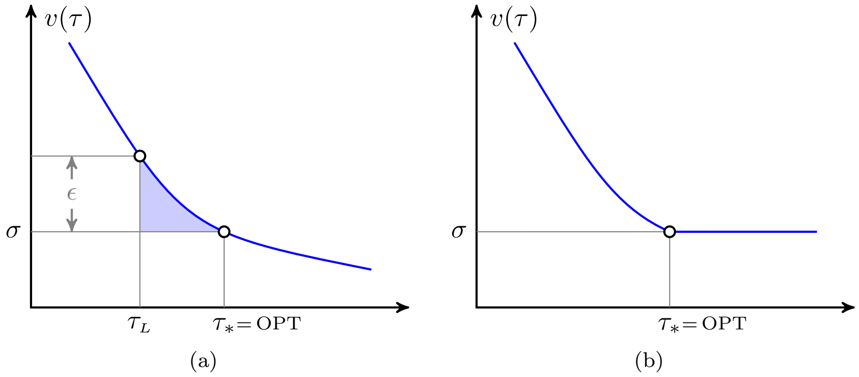


Fig. 1.1 (a) The shaded area indicates the set of allowable solutions $\tau \in [\tau_L, \tau_*]$ to the root-finding problem (1.1), and corresponds to the set of super-optimal solutions x that satisfy (1.2); (b) the root may be a minimizer of v, and then τ_* corresponds to the left-most root.

problem in terms of the value function for $\mathcal{Q}_\tau$:

$$v(\tau) := \min_{x \in \mathcal{X}} \{ \rho(Ax - b) \mid \varphi(x) \le \tau \} . \tag{1.3}$$

This value function is nonincreasing and convex [50, Theorem 5.3]. Under the mild assumption that the constraint $\rho(Ax - b) \le \sigma$ is active at any optimal solution of $\mathcal{P}_\sigma$, it is evident that the value $\tau_* :=$ OPT satisfies the equation (1.1). Conversely, it is immediate that for any $\tau \le \tau_*$ satisfying $v(\tau) \le \sigma + \epsilon$, solutions of $\mathcal{Q}_\tau$ are super-optimal and ϵ-feasible for $\mathcal{P}_\sigma$, as required. Figure 1.1(a) illustrates the set of admissible solutions.

In summary, we have translated problem $\mathcal{P}_\sigma$ to the equivalent problem of finding the minimal root of the nonlinear univariate equation (1.1). Aravkin et al. [1, Theorem 2.1] formally establish the validity of that translation. We show in §2 how approximate solutions of $\mathcal{Q}_\tau$ can serve as the basis for two Newton-like root-finding algorithms for this key equation.

In principle, any root-finding algorithm can be used. However, it must be able to obtain the left-most root, which is the only permissible solution in the case when there are multiple roots, as illustrated in Figure 1.1(b). Several algorithms are available for the root-finding problem (1.1), including bisection, secant, Newton, and their variants. These methods all require an initial estimate of the left-most root τ_*. Bisection requires two initial estimates that bracket the root, and thus it may not always be suitable in this context because it must be initialized with an upper bound on the optimal value $\tau_* =$ OPT, which may be costly to compute. (Any feasible solution of $(\mathcal{P}_\sigma)$ yields an upper bound, though obtaining it may be as costly as computing an optimal solution.) On the other hand, both secant and Newton can use initializations that underestimate the optimal value. In many important cases, this is trivial: for example, if ϕ is a norm, then $\tau = 0$ is an obvious candidate. Secant, of course, requires a second initial point, which may not be obviously available. The root-finding algorithm should also allow for inexact evaluations of the value function, and hence allow for approximate and efficient solutions of the subproblems that define (1.3). If the algorithm used to solve the subproblems is a feasible method, efficiencies may be gained if the root-finding algorithm generates iterates that are monotonically increasing, because then solutions for one subproblem are immediately feasible for the next subproblem in the sequence.

We focus on variants of secant and Newton methods that accommodate inexact oracles for v, and which enjoy an unconditional global linear rate of convergence. The secant method requires an inexact evaluation oracle that provides upper and lower bounds on v (Definition 2.1). The Newton method additionally requires a global affine minorant (Definition 2.2). Both algorithms exhibit the desirable monotonicity property described above. Coupled with an evaluation oracle for v that has a cost that is sublinear in ϵ, we obtain an algorithm with an overall cost that is also sublinear in ϵ, modulo a logarithmic factor.

1.2 Roadmap

We prove in §2 complexity bounds and convergence guarantees for the level-set scheme. We note that the iteration bounds for the root-finding schemes are independent of the slope of v at the root. This implies that the proposed method is insensitive to the "width" of the feasible region in $\mathcal{P}_\sigma$. Such methods are well-suited for problems $\mathcal{P}_\sigma$ for which the Slater constraint qualification fails or is close to failing (cf. Example 4.2). In §3, we consider refinements to the overall method, focusing on linear least-squares constraints and recovering feasibility. Section 4 explores level-set methods in notable optimization domains, including semi-definite programming, gauge optimization, and regularized regression. We also describe the specific steps needed to implement the root-finding approach for some representative applications, including low-rank matrix completion [35, 47], and sensor-network localization [7, 9, 10].

1.3 Related work

The intuition for interchanging the role of the objective and constraint functions has a distinguished history, appearing even in antiquity. Perhaps the earliest instance is Queen Dido's problem and the fabled origins of Carthage [23, Page 548]. In short, that problem is to find the maximum area that can be enclosed by an arc of fixed length and a given line. The converse problem is to find an arc of least length that traps a fixed area between a line and the arc. Although these two problems reverse the objective and the constraint, the solution in each case is a semi-circle. The interchange of constraint and objective is at the heart of Markowitz mean-variance portfolio theory [37], where the objective and constraint roles correspond to the rate of return and variance of a portfolio. The great variety of possible modern applications is formalized by the inverse function theorem in Aravkin et al. [1, Theorem 2.1]. More generally, trade-offs between various objectives form the foundations for multi-objective optimization [39].

The idea of rephrasing a constrained optimization problem as a root-finding problem has been used for at least half a century to the work of Morrison [40] and Marquadt [38]. The approach there is to minimize a quadratic function $q(x)$ subject to the trust-region constraint:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ q(x)\ \ \text{subject to}\ \ \|x\|_2 \le \varDelta.$$

Newton's method is used to compute a root for the equation $\|x(\lambda)\|_2^2 - \Delta = 0$, where $x(\lambda)$ is the solution of a parameterized unconstrained problem. This is the basis for the family of trust-region algorithms for constrained and unconstrained optimization. Newton's method for the trust-region subproblem motivated the SPGL1 algorithm [52, 53] for the 1-norm regularized least-squares problem and its extensions [1]. A shortcoming of the numerical theory to date is the absence of practical complexity and convergence guarantees. In this work, we take a fresh new look at this general framework and provide rigorous convergence guarantees. Several examples illustrate the vast applicability of the approach, and show how the proposed framework can be instantiated in concrete circumstances.

The root-finding approach is central to the ideas pioneered by Lemaréchal et al. [33], who propose a level bundle method for convex optimization [32, 56]. They consider the convex optimization problem

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad f_0(x) \quad \text{subject to} \quad f_j(x) \le 0 \text{ for } j = 1, \dots, m, \tag{1.4}$$

where each function f_j is convex and $\mathcal{X}$ is a nonempty closed convex set. The root-finding equation is based on the function

$$g(\tau) := \min_{x\in\mathcal{X}} \ \max\,\{f_0(x) - \tau,\ f_1(x), \dots,\ f_m(x)\},$$

and their algorithm constructs the smallest solution τ_* to the equation $g(\tau) = 0$, which corresponds to the optimal value (1.4). This method is also analyzed in depth by Nesterov [42, §3.3.4]

More recently, Harchaoui et al. [28] present an algorithm focusing on instances of $\mathcal{P}_\sigma$where the constraint function ρ is smooth and φ is a gauge function defined by the intersection of a unit ball for a norm and a closed convex cone. Their zero-finding method is coupled with the Frank-Wolfe algorithm, which generates lower bounds and affine minorants on the value function. In contrast, our root finding phase does not depend on the algorithm used to solve the subproblems, as is the case in the approaches described by Aravkin et al. [1] and van den Berg and Friedlander [52, 53]. In particular, the approach we take can use any affine minorant obtained from a dual certificate, as we describe in §2.3. Primal-dual algorithms can generate such certificates, but are not always practical for large-scale problems. On the other hand, first-order methods are often more suitable for large problems, but it not always obvious how to generate such certificates. Affine minorants can derived from the Frank-Wolfe algorithm (cf. §2.3), and from other families of first-order methods [20]. One approach that may be used in practice is to apply first-order methods in parallel to the the primal and dual problems.

1.4 Notation

The notation we use is standard, and closely follows that of Rockafellar [50]. For any function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$, we use the shorthand $[f \le \alpha] := \{x \mid f(x) \le \alpha\}$ to denote the α-sublevel set. An *affine minorant* of f is any affine function g satisfying $g(x) \le f(x)$ for all x. For any set $\mathcal{C} \subseteq \mathbb{R}^n$, we define the associated indicator function $\delta_{\mathcal{C}}$ vanishes over $\mathcal{C}$ and is infinite elsewhere. The p-norms and corresponding closed unit balls are denoted, respectively, by $\|\cdot\|_p$ and $\mathbb{B}_p$. For any convex cone $\mathcal{K}$ defined over a

Euclidean space, the *dual cone* is defined by $\mathcal{K}^* := \{y \mid \langle x, y\rangle \geq 0 \text{ for all } x \in \mathcal{K}\}$. The norm on that space is given by $\|x\| = \sqrt{\langle x, x\rangle}$.

We endow the space of real $m \times n$ matrices with the trace product $\langle X, Y\rangle := \operatorname{tr}(X^T Y)$ and the induced Frobenius norm $\|X\|_F := \sqrt{\langle X, X\rangle}$. The Euclidean space of real $n \times n$ symmetric matrices, written as $\mathcal{S}^n$, inherits the trace product $\langle X, Y\rangle := \operatorname{tr}(XY)$ and the corresponding norm. The closed, convex cone of $n \times n$ positive semi-definite matrices is denoted by $\mathcal{S}^n_+ = \{X \in \mathcal{S}^n \mid X \succeq 0\}$.

2 Root-finding with inexact oracles

Algorithms that provides approximate solutions of $\mathcal{Q}_\tau$ are central to our framework because these constitute the oracles through which we access v. In this section, we describe the complexity guarantees associated with two types of oracles: an inexact-evaluation oracle that provides upper and lower bounds on $v(\tau)$, and an affine-minorant oracle that additionally provides a global linear underestimator on v. For example, any primal-dual algorithm provides the required upper and lower bounds, and algorithms such as Frank-Wolfe [25, 29] automatically provide a global linear minorant, which gives approximate derivative information. The ability to use inexact solutions is crucial in practice, where the effort needed for each oracle call must be bounded.

As a counterpoint to the global complexity guarantees for inexact oracles that we describe later in this section, Theorem 2.1 describes the asymptotic superlinear rate of convergence for the secant and Newton methods with exact evaluations of any nonsmooth convex function. To our knowledge, the superlinear convergence of the secant method for convex root finding does not appear in the literature and so we provide the proof of this result in the appendix.

Theorem 2.1 (Superlinear convergence of secant and Newton methods) *Let $f\colon \mathbb{R} \to \mathbb{R}$ be a decreasing, convex function on the interval $[a, b]$. Suppose that the point $\tau_* := \inf\{\tau \mid f(\tau) \leq 0\}$ lies in (a, b) and the non-degeneracy condition $g_* := \inf\{g \mid g \in \partial f(\tau_*)\} < 0$ holds. Fix two points $\tau_0, \tau_1 \in (a, b)$ satisfying $\tau_0 < \tau_1 < \tau_*$ and consider the following two iterations:*

$$\tau_{k+1} := \begin{cases} \tau_k & \text{if } f(\tau_k) = 0, \\ \tau_k - \frac{f(\tau_k)}{g_k} \quad [\text{with } g_k \in \partial f(\tau_k)] & \text{otherwise;} \end{cases} \tag{Newton}$$

and

$$\tau_{k+1} := \begin{cases} \tau_k & \text{if } f(\tau_k) = 0, \\ \tau_k - \frac{\tau_k - \tau_{k-1}}{f(\tau_k) - f(\tau_{k-1})} f(\tau_k) & \text{otherwise.} \end{cases} \tag{Secant}$$

If either sequence terminates finitely at some τ_k, then it must be the case $\tau_k = \tau_*$. If the sequence $\{\tau_k\}$ does not terminate finitely, then $|\tau_* - \tau_{k+1}| \leq (1 - g_*/\gamma_k)|\tau_* - \tau_k|$, $k = 1, 2, \ldots$, where $\gamma_k = g_k$ for the Newton sequence and γ_k is any element of $\partial f(\tau_{k-1})$ for the secant sequence. In either case, $\gamma_k \uparrow g_*$ and $\tau_k \uparrow \tau_*$ globally q-superlinearly.

The algorithms presented here apply to any convex decreasing function $f : \mathbb{R} \to \mathbb{R}$ for which the equation $f(\tau) = 0$ has a solution. In the following discussion, τ_* denotes a minimal root of $f(\tau) = 0$. Given a tolerance $\epsilon > 0$, the algorithms we discuss yield a point $\tau \leq \tau_*$ satisfying $0 \leq f(\tau) \leq \epsilon$.

Algorithm 2.1: Inexact secant method

```
Data: Target accuracy $\epsilon > 0$; a decreasing convex function $f : \mathbb{R} \to \mathbb{R}$ via an inexact
      evaluation oracle $\mathcal{O}_{f,\epsilon}$; initial points $\tau_0, \tau_1$ with $\tau_0 < \tau_1$ such that $f(\tau_1) > 0$;
      constant $\alpha \in (1, 2)$.
$(\ell_0, u_0) \leftarrow \mathcal{O}_{f,\epsilon}(\tau_0, \alpha)$,   $(\ell_1, u_1) \leftarrow \mathcal{O}_{f,\epsilon}(\tau_1, \alpha)$,   $u_1 \leftarrow \min(u_1, u_0)$,   $k \leftarrow 1$
while $u_k > \epsilon$ do
    $s_k \leftarrow (u_{k-1} - \ell_k)/(\tau_{k-1} - \tau_k)$          [slope of linear approximation]
    $\tau_{k+1} \leftarrow \tau_k - \ell_k/s_k$                        [secant iteration]
    $(\ell_{k+1}, u_{k+1}) \leftarrow \mathcal{O}_{f,\epsilon}(\tau_{k+1}, \alpha)$   [oracle evaluation for lower/upper bounds]
    $u_{k+1} \leftarrow \min\{u_{k+1}, u_k\}$                           [ensure upper bound decreases]
    $k \leftarrow k + 1$
return $\tau_k$
```

2.1 Inexact secant

Our first root-finding algorithm is an inexact secant method, and is based on an oracle that provides upper and lower bounds on the value $f(\tau)$.

Definition 2.1 (Inexact evaluation oracle) For a function $f : \mathbb{R} \to \mathbb{R}$ and $\epsilon \geq 0$, an *inexact evaluation oracle* is a map $\mathcal{O}_{f,\epsilon}$ that assigns to each pair $(\tau, \alpha) \in [f > 0] \times [1, \infty)$ real numbers (ℓ, u) such that $\ell \leq f(\tau) \leq u$ and either $u \leq \epsilon$, or $u > \epsilon$ and $1 \leq u/\ell \leq \alpha$.

This oracle guarantees that either (i) we obtained an ϵ-accurate solution τ, or (ii) that $\ell > 0$ and $1 \leq u/\ell \leq \alpha$. The ratio u/ℓ measures the relative optimality of the point τ. In contrast to the absolute gap $u - \ell$, it allows the oracle to be increasingly inexact for larger values of $f(\tau)$. The relative-accuracy condition is no less general than one based on an absolute gap. In particular, we can verify that if the absolute gap satisfies the condition $u - \ell \leq (1 - 1/\alpha)\epsilon$, then u and ℓ satisfy the conditions required by the inexact evaluation oracle. Indeed, provided that $u > \epsilon$, we deduce $u/\ell \leq 1 + (1 - 1/\alpha)\epsilon/\ell \leq 1 + (1 - 1/\alpha)u/\ell$. After rearranging terms, this yields the desired inequality $1 \leq u/\ell \leq \alpha$.

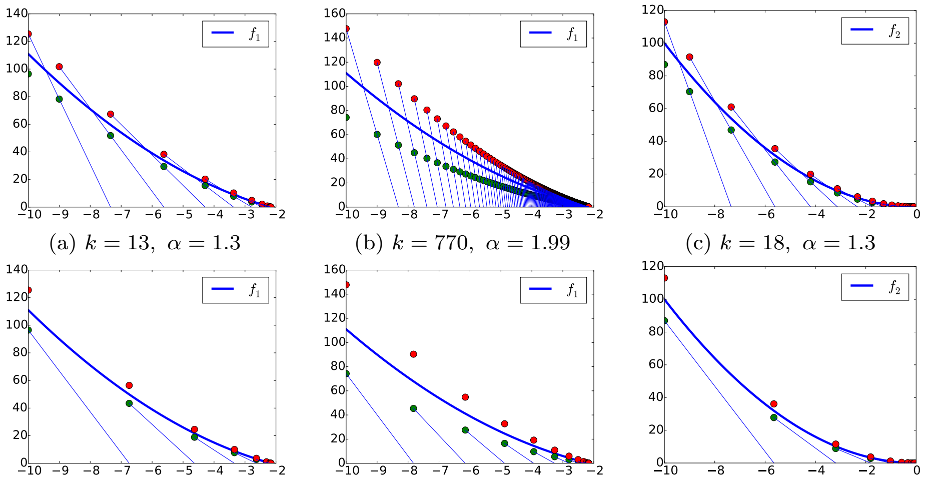
Algorithm 2.1 outlines a secant method based on the inexact evaluation oracle. Theorem 2.2 establishes the corresponding global convergence guarantees; the proof appears in Appendix A.

Theorem 2.2 (Linear convergence of the inexact secant method) *The inexact secant method (Algorithm 2.1) terminates after at most*

$$k \leq \max\left\{2 + \log_{2/\alpha}(2C/\epsilon)\,,\, 3\right\}$$

iterations, where $C := \max\{|s_1|(\tau_* - \tau_1),\, \ell_1\}$ *and* $s_1 := (u_0 - \ell_1)/(\tau_0 - \tau_1)$.

The iteration bound of the inexact secant method is indifferent to the slope of the function f at the minimal root τ_* because termination depends on function values rather than proximity to τ_*. The plots in Figure 2.1 illustrate this behavior: panel (a) shows the iterates for $f_1(\tau) = (\tau - 1)^2 - 10$, which has a nonzero slope at the minimal root $\tau_* = 1 - \sqrt{10} \approx -2.2$ and so has a non-degenerate solution; panel (c) shows the iterates for $f_2(\tau) = \tau^2$, which is clearly degenerate at the solution. The algorithm behaves similarly on both problems. When applied to the value function



(a) $k = 13,\ \alpha = 1.3$ (b) $k = 770,\ \alpha = 1.99$ (c) $k = 18,\ \alpha = 1.3$

(d) $k = 9,\ \alpha = 1.3$ (e) $k = 15,\ \alpha = 1.99$ (f) $k = 10,\ \alpha = 1.3$

Fig. 2.1 Inexact secant method (top row) and Newton method (bottom row) for root finding on the functions $f_1(\tau) = (\tau - 1)^2 - 10$ (first two columns) and $f_2(\tau) = \tau^2$ (last column). Below each panel, α is the oracle accuracy, and k is the number of iterations needed to converge, i.e., to reach $f_i(\tau_k) \le \epsilon$. For all problems, $\epsilon = 10^{-2}$; the horizontal axis is τ, and the vertical axis is $f_i(\tau)$.

v to find a root of (1.1), the algorithm's indifference to degeneracy translates to an insensitivity to the width [48] of the feasible region of $\mathcal{P}_\sigma$ —a consequence of the fact that the scheme maintains infeasible iterates for $\mathcal{P}_\sigma$. Such methods are well-suited for problems $\mathcal{P}_\sigma$ for which the Slater constraint qualification is close to failing.

The iteration bound in Theorem 2.2 is infinite for $\alpha \ge 2$. This is not an artifact of the proof. As illustrated by Figure 2.1(b), the inexact secant method behaves poorly for α close to 2. Indeed, it can fail to converge linearly (or at all) to the minimal root for any $\alpha \ge 2$, as the following example shows. Consider the linear function $f(\tau) = -\tau$ with lower and upper bounds $\ell_k := -2\tau_k/(1+\alpha)$ and $u_k := -2\alpha\tau_k/(1+\alpha)$. A quick computation shows that the quotients $q_k := \tau_k/\tau_{k-1}$ of the iterates satisfy the recurrence relation $q_{k+1} = (1-\alpha)/(q_k - \alpha)$. It is then immediate that for all $\alpha \ge 2$, the quotients q_k tend to one, indicating that the method stalls.

2.2 Inexact Newton

The secant method can be improved by using approximate derivative information (when available) to design a Newton-type method. We design an inexact Newton method around an improved oracle that provides global linear under-estimators of f. This approach has two main advantages over the secant method. First, it is guaranteed to take longer steps than the inexact secant method. Second, it locally converges quadratically whenever f is smooth, the values $f(\tau)$ are computed exactly, and the function has a nonzero (left) derivative at the minimal root. To

Algorithm 2.2: Inexact Newton method

```
Data: Target accuracy $\epsilon > 0$; convex decreasing function $f\colon \mathbb{R} \to \mathbb{R}$ via an affine
      minorant oracle $\mathcal{O}_{f,\epsilon}$; initial point $\tau_0$ with $f(\tau_0) > 0$; constant $\alpha \in (1, 2)$.
$u_{-1} \leftarrow +\infty$,   $(\ell_0, u_0, s_0) \leftarrow \mathcal{O}_{f,\epsilon}(\tau_0, \alpha)$,   $k \leftarrow 0$
while $u_k > \epsilon$ do
    $\tau_{k+1} \leftarrow \tau_k - \ell_k/s_k$                                  [Newton iteration]
    $(\ell_{k+1}, u_{k+1}, s_{k+1}) \leftarrow \mathcal{O}_{f,\epsilon}(\tau_k, \alpha)$     [evaluate lower affine minorant oracle]
    $u_k \leftarrow \min\{u_k, u_{k-1}\}$                           [ensure upper bound decreases]
    $k \leftarrow k + 1$
return $\tau_k$
```

formalize these ideas, we use the following strengthened version of an inexact evaluation oracle.

Definition 2.2 (Affine minorant oracle) For a function $f : \mathbb{R} \to \mathbb{R}$ and $\epsilon \geq 0$, an *affine minorant oracle* is a mapping $\mathcal{O}_{f,\epsilon}$ that assigns to each pair $(\tau, \alpha) \in [f > 0] \times [1, \infty)$ real numbers (ℓ, u, s) such that $\ell \leq f(\tau) \leq u$, and either $u \leq \epsilon$, or $u > \epsilon$ and $1 \leq u/\ell \leq \alpha$. The affine function $\bar{\tau} \mapsto \ell + s(\bar{\tau} - \tau)$ globally minorizes f.

Algorithm 2.2 gives a Newton method based on the affine minorant oracle. The inexact Newton method has global convergence guarantees analogous to those of the inexact secant method, as described in Theorem 2.3. A proof is given in Appendix A.

Theorem 2.3 (Linear convergence of the inexact Newton method) *The inexact Newton method (Algorithm 2.2) terminates after at most*

$$k \leq \max\left\{1 + \log_{2/\alpha}(2C/\epsilon)\,,\, 2\right\}$$

iterations, where $C := \max\{|s_0|(\tau_* - \tau_0),\, \ell_0\}$.

When we compare the two algorithms, it is easy to see that the Newton steps are never shorter than the secant steps. Indeed, let $(\ell_{k-1}, u_{k-1}, s_{k-1}) = \mathcal{O}_f(\tau_{k-1}, \alpha)$ and $(\ell_k, u_k, s_k) = \mathcal{O}_f(\tau_k, \alpha)$ be the triples returned by an affine minorant oracle at τ_{k-1} and τ_k, respectively. Then

$$u_{k-1} \geq f(\tau_{k-1}) \geq \ell_k + s_k(\tau_{k-1} - \tau_k),$$

which implies

$$s_k^{\text{secant}} := (u_{k-1} - \ell_k)/(\tau_{k-1} - \tau_k) \leq s_k =: s_k^{\text{newton}}.$$

Therefore, the Newton step length $-\ell_k/s_k^{\text{newton}}$ is at least as large as the secant step length $-\ell_k/s_k^{\text{secant}}$.

The Newton method often outperforms the secant method in practice. The bottom row of panels in Figure 2.1 shows the progress of the Newton method on the same test problems specfied earlier. The Newton method performs relatively well even when α is near its upper limit of 2; compare panels (b) and (e) in the figure. In this set of experiments, we chose an oracle that has the same quality lower and upper bounds as the experiments with secant, but has the least favorable (i.e., steepest) slope that still results in a global minorant.

2.3 Lower minorants via duality

When are affine minorant oracles of the value function v readily available? Suppose we can express the value function in *dual form*,

$$v(\tau) = \max_{y \in \mathbb{R}^m} \ \Phi(y, \tau),$$

where Φ is concave in y and convex in τ. For example, appealing to Fenchel duality, we may write

$$\begin{aligned} v(\tau) &= \min_{x \in \mathcal{X}} \ \{\rho(Ax - b) \mid \varphi(x) \le \tau\} \\ &= \min_{x \in \mathbb{R}^n} \ \rho(Ax - b) + \delta_{\mathcal{X} \cap \, [\varphi \le \tau]}(x) \\ &= \max_{y \in \mathbb{R}^m} \ \langle y, b \rangle - \rho^\star(-y) - \delta^\star_{\mathcal{X} \cap \, [\varphi \le \tau]}(A^* y), \end{aligned}$$

where the last equality holds provided that either the primal or the dual problem has a strictly feasible point [11, Theorem 3.3.5]. Hence, the Fenchel dual objective

$$\Phi(y, \tau) := \langle b, y \rangle - \rho^\star(-y) - \delta^\star_{\mathcal{X} \cap [\varphi \le \tau]}(A^* y) \tag{2.1}$$

yields an explicit representation for Φ, which is concave in τ, as shown by Lemma A.1.

Many standard first-order methods that might be used as an oracle for evaluating $v(\bar{\tau}) - \sigma$, and generate both a lower bound $\bar{\ell}$ and a dual certificate $\bar{y}$ that satisfy the equation $\bar{\ell} = \Phi(\bar{y}, \bar{\tau}) - \sigma$. Examples include saddle-prox [41], Frank-Wolfe [25, 29], some projected subgradient methods [2], and accelerated versions [43, 51]. Whenever such a dual certificate $\bar{y}$ is available, we have

$$\begin{aligned} v(\tau) - \sigma \ge \Phi(\bar{y}, \tau) - \sigma &= \big(\Phi(\bar{y}, \bar{\tau}) - \sigma\big) + \big(\Phi(\bar{y}, \tau) - \Phi(\bar{y}, \bar{\tau})\big) \\ &\ge \bar{\ell} + \bar{s}(\tau - \bar{\tau}), \end{aligned} \tag{2.2}$$

where $\bar{s} \in \partial_\tau \delta^\star_{\mathcal{X} \cap \, [\varphi \le \tau]}(A^* y)$. Hence, any dual certificate $\bar{y}$ is valid if it generates an affine minorant oracle where $\bar{\ell}$ satisfies the accuracy condition required by Definition 2.2. In summary, if y_k is a valid dual certificate, we may take

$$\ell_k := \Phi(y_k, \tau_k) - \sigma \quad \text{and any} \quad s_k \in \partial_\tau \delta^\star_{\mathcal{X} \cap \, [\varphi \le \tau_k]}(A^* y_k).$$

We derive in §4 subdifferential formulas for a large class of contemporary problems, including conic and gauge optimization (cf. Tables 4.1 and 4.2). More general rules for computing subdifferential formulas are outlined by Aravkin et al. [1, Equations 5.1(d,e), 6.13, 6.26].

2.4 Lower minorants via Frank-Wolfe

In some instances, lower-bounds on the optimal value of $\mathcal{Q}_\tau$ that is provided by an algorithm are seemingly not related to a dual solution. A notable example of such a scheme is the Frank-Wolfe algorithm, which has recently received much attention.

Suppose that the function ρ is smooth. The Frank-Wolfe method applied to the problem $\mathcal{Q}_\tau$ is based on the following two-step iteration:

$$\begin{aligned} z_k &= \operatorname{argmin}\ \left\{\langle A^*\nabla\rho(Ax_k - b), z\rangle \mid z \in \mathcal{X} \cap [\varphi \leq \tau]\right\} \\ x_{k+1} &= x_k + t_k(z_k - x_k) \end{aligned}$$

for an appropriately chosen sequence of step-sizes t_k (e.g., $t_k = \frac{2}{k+2}$). As the method progresses, it generates the upper bounds $u_k = \min_{i=1,\dots,k} \rho(Ax_i - b)$ on the optimal value of $\mathcal{Q}_\tau$. Moreover, it is easy to deduce from convexity that the following are valid lower bounds:

$$\ell_k = \max_{i=1,\dots,k} \left\{\rho(Ax_i - b) + \langle A^*\nabla\rho(Ax_i - b), z_i - x_i\rangle\right\}.$$

Jaggi [29] provides an extensive discussion. If the step sizes t_k are chosen appropriately, the gap satisfies $u_k - \ell_k \leq \mathcal{O}(D^2 L/k)$, where the diameter D of the feasible region and the Lipschitz constant L of the gradient of the objective function of $\mathcal{Q}_\tau$ are measured in an arbitrary norm. Harchaoui et al. [28] observe how to deduce from such lower bounds ℓ_k an affine minorant of the value function v.

On the other hand, one can also show that the lower bounds ℓ_k are indeed generated by an explicit candidate dual solution, and hence the Frank-Wolfe algorithm (and its variants) fit perfectly in the above framework based on dual certificates. To see this, consider the Fenchel dual

$$\underset{y\in\mathbb{R}^m}{\text{maximize}} \quad \Phi(y,\tau) = \langle y, b\rangle - \rho^\star(-y) - \delta^\star_{\mathcal{X}\cap[\varphi\leq\tau]}(A^*y)$$

of $\mathcal{Q}_\tau$. Then for the candidate dual solutions $y_i := -\nabla\rho(Ax_i - b)$, we deduce

$$\begin{aligned} \Phi(y_i,\tau) &= \langle y_i, b\rangle - \rho^\star(-y_i) - \langle A^*y_i, z_i\rangle \\ &= \langle y_i, b\rangle + \Big(\rho(Ax_i - b) + \langle y_i, Ax_i - b\rangle\Big) - \langle A^*y_i, z_i\rangle \\ &= \rho(Ax_i - b) + \langle A^T\nabla\rho(Ax_i - b), z_i - x_i\rangle. \end{aligned}$$

Thus, the lower bounds ℓ_k are simply equal to $\ell_k = \max_{i=1,\dots,k} \Phi(y_i,\tau)$, and affine minorants on the value function v are readily computed from the dual iterates y_k and the derivatives $\partial_\tau\delta^\star_{\mathcal{X}\cap\,[\varphi\leq\tau]}(A^*y_k)$.

3 Special cases

This section can be considered as an aside in our main exposition. We address in this section two questions that arise in the application of our root-finding approach: how best to apply the algorithm to problems with linear least-squares constraints, and how to recover a feasible point.

3.1 Least-squares misfit and degeneracy

Particularly important instances of problem $\mathcal{P}_\sigma$ arise when the misfit between Ax and b is measured by the 2-norm, i.e., $\rho = \|\cdot\|_2$. In this case, the objective of the level-set problem $\mathcal{Q}_\tau$ is $\|Ax-b\|_2$, which is not differentiable whenever $Ax = b$. Rather than applying a nonsmooth optimization scheme, an apparently easy fix is to replace the constraint in $\mathcal{P}_\sigma$ with its equivalent formulation $\frac{1}{2}\|Ax-b\|_2^2 \le \frac{1}{2}\sigma^2$, leading to the pair of problems

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad \varphi(x) \qquad \text{subject to} \quad \tfrac{1}{2}\|Ax-b\|_2^2 \le \tfrac{1}{2}\sigma^2, \tag{$\mathcal{P}_\sigma^2$}$$

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad \tfrac{1}{2}\|Ax-b\|_2^2 \quad \text{subject to} \quad \varphi(x) \le \tau. \tag{$\mathcal{Q}_\tau^2$}$$

However, this reformulation presents some numerical difficulties. Below we describe the potential pitfalls and a simple alternative.

For this section only, define

$$f_1(\tau) := v(\tau) - \sigma \qquad \text{and} \qquad f_2(\tau) := \tfrac{1}{2}v^2(\tau) - \tfrac{1}{2}\sigma^2,$$

where v is the value function corresponding to the original (unsquared) level-set problem $\mathcal{Q}_\tau$. Throughout this section, the problems $\mathcal{P}_\sigma$ and $\mathcal{Q}_\tau$ continue to define the original formulations without the squares.

A direct application of the root-finding procedure for $\mathcal{P}_\sigma^2$ would be applied to the function f_2, which is degenerate at each of its roots. As a result, the secant and Newton root-finding methods would not converge locally superlinearly—even if the values $v(\tau)$ are evaluated exactly; see Theorem 2.1. Moreover, we have observed empirically that this issue can in some cases cause numerical schemes to stagnate.

A simple alternative avoids this pitfall: apply the root-finding procedure to the function f_1, but approximately solve $\mathcal{Q}_\tau^2$ to obtain bounds on v. The oracle definitions required for the secant (Algorithm 2.1) and Newton (Algorithm 2.2) methods require suitable modification. For secant, the modifications are straightforward, but for Newton, care is needed in order to obtain the correct affine minorants of f_1. The required modifications for secant and Newton are described below.

Secant. For the secant method applied to the function f_1, we derive an inexact evaluation oracle from an inexact evaluation oracle for f_2 as follows. Suppose that we have approximately solved $\mathcal{Q}_\tau^2$ by an inexact-evaluation oracle

$$\mathcal{O}_{f_2,\epsilon}(\tau,\alpha^2) = \left(\tfrac{1}{2}\ell^2 - \tfrac{1}{2}\sigma^2,\ \tfrac{1}{2}u^2 - \tfrac{1}{2}\sigma^2\right), \tag{3.1}$$

where we have specified the relative accuracy between the lower and upper bounds to be α^2. Assume, without loss of generality, that $u, \ell \ge 0$. Then clearly u and ℓ are upper and lower bounds on $v(\tau)$, respectively. It is now straightforward to deduce

$$0 \le \ell - \sigma \le f_1(\tau) \le u - \sigma \qquad \text{and} \qquad \frac{u-\sigma}{\ell-\sigma} \le \sqrt{\frac{u^2-\sigma^2}{\ell^2-\sigma^2}} \le \alpha. \tag{3.2}$$

Hence an inexact function evaluation oracle for f_2 yields an inexact evaluation oracle for f_1.

Newton. Newton's method in this setting is slightly more intricate because of the formulas required for obtaining a valid affine minorant of f_1. We use the respective objectives of the dual problems corresponding to $\mathcal{Q}_\tau$ and $\mathcal{Q}_\tau^2$, given by

$$\Phi_1(y,\tau) := \langle b, y\rangle - \delta^\star_{\mathcal{X}\cap[\varphi\le\tau]}(A^*y) - \delta_{\mathbb{B}_2}(y),$$
$$\Phi_2(y,\tau) := \langle b, y\rangle - \delta^\star_{\mathcal{X}\cap[\varphi\le\tau]}(A^*y) - \tfrac{1}{2}\|y\|_2^2.$$

As described by (3.1), an inexact solution of $\mathcal{Q}_\tau^2$ delivers values ℓ and u that satisfy (3.2). Let y be the valid dual certificate that generated the lower bound ℓ, so that $\Phi_2(y,\tau) = \frac{1}{2}\ell^2$. (See the discussion in §2.3 regarding valid dual certificates.) Let $s \in \partial_\tau\Phi_2(y,\tau)$ be any subgradient. The following result establishes that $(\hat\ell,\, u,\, s/\|y\|_2)$, with $\hat\ell := \Phi_1\left(y/\|y\|_2,\, \tau\right)$, defines a valid affine minorant for f_1.

Proposition 3.1 *The inequalities*

$$0 \le \hat\ell - \sigma \le f_1(\tau) \le u - \sigma \qquad \text{and} \qquad (u-\sigma)/(\hat\ell - \sigma) \le \alpha$$

hold, and the linear functional $\tau' \mapsto (\hat\ell - \sigma) - (s/\|y\|_2)(\tau' - \tau)$ *minorizes* f_1.

The proof is given in Appendix A. In summary, if we wish to obtain a super-optimal and ϵ-feasible solution to $\mathcal{P}_\sigma$, in each iteration of the Newton method we must evaluate $f_2(\tau)$ up to an absolute error of at most $\frac{1}{2}(1-1/\alpha)^2\epsilon^2$. Indeed, suppose that in the process of evaluation, the oracle $\mathcal{O}_{f_2}\left(\tau, \alpha^2\right)$ achieves u and l satisfying $\frac{1}{2}u^2 - \frac{1}{2}\ell^2 \le \frac{1}{2}(1-1/\alpha)^2\epsilon^2$. Then we obtain the inequality

$$u - \ell = \sqrt{(u-\ell)^2} \le \sqrt{u^2 - \ell^2} \le (1-1/\alpha)\epsilon.$$

Thus, by the discussion following Definition 2.1, either the whole Newton scheme can now terminate with $f_1(\tau) \le \epsilon$ or we have achieved the relative accuracy $(u-\sigma)/(\ell-\sigma) \le \alpha$ for the oracle.

4 Some problem classes

A wide variety of problems can be treated by the root-finding approach, including sparse optimization, with applications in compressed sensing and sparse recovery, and conic optimization, including semidefinite programming (SDP). The following sections give recipes for applying the root-finding approach in different contexts.

4.1 Conic optimization

The general conic problem (CP) has the form

$$\underset{x}{\text{minimize}} \quad \langle c, x\rangle \quad \text{subject to} \quad \mathcal{A}x = b,\ x \in \mathcal{K}, \tag{CP}$$

where $\mathcal{A} : E_1 \to E_2$ is a linear map between Euclidean spaces, and $\mathcal{K} \subset E_1$ is a proper, closed, convex cone. The familiar forms of this problem include linear programming (LP), second-order cone programming (SOCP), and semidefinite programming (SDP). Ben-Tal and Nemirovski [5] survey an enormous number of applications and formulations captured by conic programming.

There are at least two possible approaches for applying the level-set framework to this problem. The first approach exchanges the roles of the original objective $\langle c, x\rangle$ with the linear constraint $Ax = b$, and brings a least-squares term into the objective. The second approach moves the cone constraint $x \in \mathcal{K}$ into the objective with the aid of a suitable distance function. This yields two distinct algorithms for the conic problem. The two approaches are summarized in Table 4.1. Note that it is possible to consider conic problems with the more general constraint $\rho(Ax - b) \le \sigma$, but here we restrict our attention to the simpler affine constraint, which conforms to the standard form of conic optimization.

4.1.1 First approach: least-squares level set

In this section we describe an application of the level-set approach to (CP) that exchanges the roles of the linear functions, and derive the overall complexity guarantees. This approach relies on a simple transformation that guarantees a lower bound on the objective value. To this end, we make the blanket assumption that there is available a *strictly feasible* vector $\widehat{y}$ the dual of (CP)

$$\underset{y}{\text{maximize}} \quad \langle b, y\rangle \quad \text{subject to} \quad c - \mathcal{A}^* y \in \mathcal{K}^*.$$

Thus $\widehat{y}$ satisfies $\widehat{c} := c - \mathcal{A}^*\widehat{y} \in \operatorname{int}\mathcal{K}^*$. A simple calculation shows that minimizing the new objective $\langle \widehat{c}, x\rangle$ only changes the objective of CP by a constant: for all x feasible for CP, we now have

$$\langle \widehat{c}, x\rangle = \langle c, x\rangle - \langle \mathcal{A}x, \widehat{y}\rangle = \langle c, x\rangle - \langle b, \widehat{y}\rangle.$$

In particular, we may assume $b \neq 0$, since otherwise, the origin is the trivial solution for the shifted problem. Note that in the important case $c \in \operatorname{int}\mathcal{K}$, we can simply set $\widehat{y} = 0$, which yields the equality $c = \widehat{c}$.

We now illustrate the computational complexity of applying the root-finding approach to solve (CP) using the level-set problem

$$\underset{x}{\text{minimize}} \quad \|\mathcal{A}x - b\|_2 \quad \text{subject to} \quad \langle \widehat{c}, x\rangle \le \tau,\ x \in \mathcal{K}. \tag{4.1}$$

Our aim is then to find a root of (1.1), where v is the value function of (4.1). The top row of Table 4.1, gives the corresponding dual

$$\underset{y,\ \mu\ge 0}{\text{maximize}} \quad \langle b, y\rangle - \mu\tau \quad \text{subject to} \quad \|y\|_2 \le 1,\ \mu c - \mathcal{A}^* y \in \mathcal{K}^*$$

of the level-set problem. We use $\tau_0 = 0$ as the initial root-finding iterate. Because of the inclusion $\widehat{c} \in \operatorname{int}\mathcal{K}^*$, we deduce that $x = 0$ is the only feasible solution to (4.1), which yields $v(0) = \|b\|_2$ and the exact lower bound $\ell_0 = \|b\|_2$. The corresponding dual certificate is $(\bar{y}, \bar{\mu}) = (b/\|b\|_2, \bar{\mu})$, where

$$\bar{\mu} := \min_{\mu} \left\{ \mu\widehat{c} - (\mathcal{A}^* b)/\|b\|_2 \in \mathcal{K}^* \right\}. \tag{4.2}$$

Note the inequality $\bar{\mu} > 0$, because otherwise we would deduce $\mathcal{A}^* b \in -\mathcal{K}^*$, implying the inequality $\|b\|_2^2 = \langle b, \mathcal{A}x\rangle = \langle \mathcal{A}^* b, x\rangle \le 0$ for any feasible x. This

Problem	$\mathcal{P}_\sigma$	$\mathcal{Q}_\tau$	Dual of $\mathcal{Q}_\tau$
CP least-squares level	$\min\limits_{x}\ \langle c, x\rangle$ $\text{s.t.}\ \ \mathcal{A}x = b$ $x \in \mathcal{K}$	$\min\limits_{x}\ \Vert\mathcal{A}x - b\Vert_2$ $\text{s.t.}\ \ \langle c, x\rangle \le \tau$ $x \in \mathcal{K}$	$\max\limits_{y,\ \mu\ge 0}\ \langle b, y\rangle - \mu\tau$ $\text{s.t.}\ \ \Vert y\Vert_2 \le 1$ $\mu c - \mathcal{A}^* y \in \mathcal{K}^*$
CP cone level	$\min\limits_{x}\ \langle c, x\rangle$ $\text{s.t.}\ \ \mathcal{A}x = b$ $x \in \mathcal{K}$	$\min\limits_{x}\ -\lambda_{\min}(x)$ $\text{s.t.}\ \ \mathcal{A}x = b$ $\langle c, x\rangle \le \tau$	$\max\limits_{y,\ \mu\ge 0}\ \langle b, y\rangle - \mu\tau$ $\text{s.t.}\ \ \langle \mu c - \mathcal{A}^* y, e\rangle = 1$ $\mu c - \mathcal{A}^* y \in \mathcal{K}^*$

Table 4.1 Least-squares and conic level-set problems for conic optimization. In these examples, we require $\mathcal{A}x = b$.

contradicts our assumption that b is nonzero. In the case where $\mathcal{K}$ is the nonnegative orthant and $\widehat{c} = e$, the number $\bar{\mu}$ is simply the maximal coordinate of $\mathcal{A}^* b/\Vert b\Vert_2$; if $\mathcal{K}$ is the semidefinite cone and $\widehat{c} = I$, the number $\bar{\mu}$ is the maximal eigenvalue of $\mathcal{A}^* b/\Vert b\Vert_2$.

Let OPT denote the optimal value of CP. Theorem 2.3 asserts that within $\mathcal{O}\big(\log_{2/\alpha} 2C/\epsilon\big)$ inexact Newton iterations, where α is the accuracy of each subproblem solve and

$$C = \max\left\{\bar{\mu}\cdot(\text{OPT} - \langle b, \widehat{y}\rangle),\ \Vert b\Vert_2\right\},$$

the point $x \in \mathcal{K}$ that yields the final upper bound in (4.1) is a super-optimal and ϵ-feasible solution of the shifted CP. Thus, x satisfies

$$\langle \widehat{c}, x\rangle \le \text{OPT} - \langle \widehat{y}, b\rangle \quad \text{and} \quad \Vert\mathcal{A}x - b\Vert_2 \le \epsilon.$$

To see how good the obtained point x is for the original CP (without the shift), note that

$$\langle \widehat{c}, x\rangle = \langle c, x\rangle - \langle \mathcal{A}^*\widehat{y}, x\rangle = \langle c, x\rangle - \langle \widehat{y}, \mathcal{A}x - b\rangle - \langle \widehat{y}, b\rangle \ge \langle c, x\rangle - \langle \widehat{y}, b\rangle - \epsilon\Vert\widehat{y}\Vert_2,$$

and hence $\langle c, x\rangle \le \text{OPT} + \epsilon\Vert\widehat{y}\Vert_2$. In the important case where $c \in \operatorname{int}\mathcal{K}^*$, we deduce super-optimality $\langle c, x\rangle \le \text{OPT}$ for the target problem CP.

Each Newton root-finding iteration requires an approximate solution of (4.1). As described in §3.1, we obtain this approximation by instead solving its smooth formulation with the squared objective $(1/2)\Vert\mathcal{A}x - b\Vert_2^2$. Let $L := \Vert\mathcal{A}\Vert_2^2$, where $\Vert\mathcal{A}\Vert_2$ is the operator norm induced by the Euclidean norms on the spaces E_1 and E_2, be the Lipschitz constant for the gradient $\mathcal{A}^*(\mathcal{A}\cdot - b)$. Also, let D be the diameter of the region $\{x \mid \langle \widehat{c}, x\rangle = 1,\ x \in \mathcal{K}\}$, which is finite by the inclusion $\widehat{c} \in \operatorname{int}\mathcal{K}^*$. Thus, in order to evaluate v to an accuracy ϵ, we may apply an accelerated projected-gradient method on the squared version of the problem to an additive error of $\frac{1}{2}(1 - 1/\alpha)^2\epsilon^2$ (see end of §3.1), which terminates in at most

$$\mathcal{O}\left(\frac{\sqrt{L}\cdot\tau D}{\epsilon(1 - 1/\alpha)}\right) = \mathcal{O}\left(\frac{\Vert\mathcal{A}\Vert_2\cdot D\cdot(\text{OPT} - \langle b, \widehat{y}\rangle)}{\epsilon(1 - 1/\alpha)}\right)$$

iterations [6, §6.2]. Here, we have used the monotonicity of the root finding scheme to conclude $\tau \le \text{OPT} - \langle b, \widehat{y}\rangle$. When $\mathcal{K}$ is the non-negative orthant, each projection can be accomplished with $\mathcal{O}(n)$ floating point operations [13], while for the semidefinite

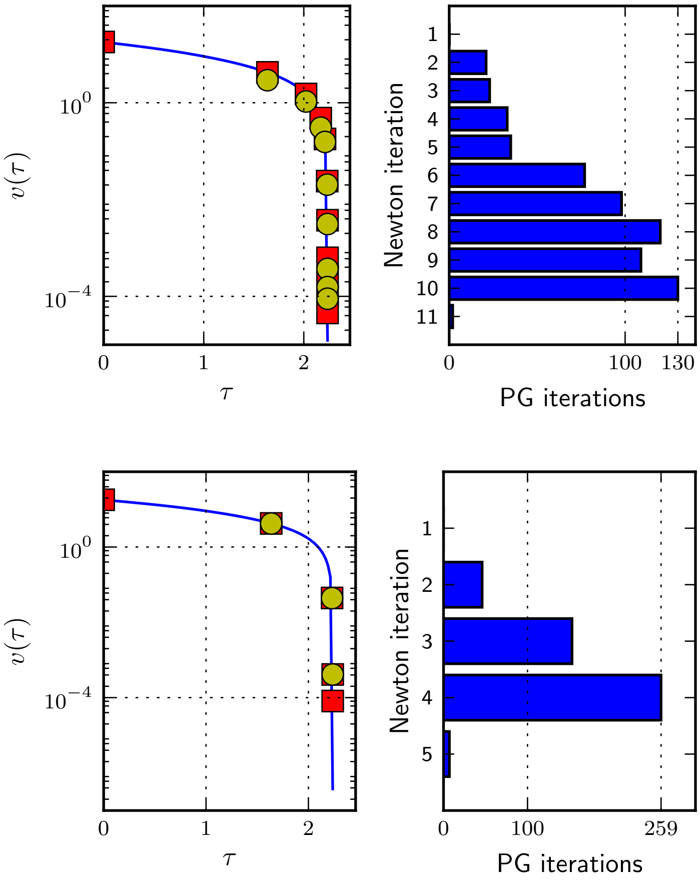


Fig. 4.1 Progress of the least-squares level-set method for a linear program (cf. §4.1.1). The panels on the left depict the graph of $v(\tau)$ (solid line), and the squares and circles, respectively, show the upper and lower bounds computed using an optimal projected-gradient method. The horizontal log scale results in a value function that appears nonconvex. The panels on the right show the number of projected-gradient iterations for each Newton step. Top panels: $\alpha = 1.8$. Bottom panels: $\alpha = 1.01$.

cone each projection requires an eigenvalue decomposition. More generally, such projections can be quickly found as long as projections onto the cone $\mathcal{K}$ are available; see Remark A.1. An improved complexity bound can be obtained for the oracles in the LP and SDP cases by replacing the Euclidean projection step with a Bregman projection derived from the entropy function; see e.g., Beck and Teboulle [4] or Tseng [51, §3.1]. We leave the details to the reader.

In summary, we can obtain a point $x \in \mathcal{K}$ that satisfies $\langle c, x\rangle \le \mathrm{OPT} + \epsilon\|\widehat{y}\|_2$ and $\|\mathcal{A}x - b\|_2 \le \epsilon$ in at most

$$\mathcal{O}\left(\frac{\|A\|_2 \cdot D \cdot (\mathrm{OPT} - \langle b, \widehat{y}\rangle)}{\epsilon(1 - 1/\alpha)}\right) \cdot \mathcal{O}\left(\log_{2/\alpha} \frac{\max\left\{\bar{\mu} \cdot (\mathrm{OPT} - \langle b, \widehat{y}\rangle), \|b\|_2\right\}}{\epsilon}\right)$$

iterations of an accelerated projected-gradient method, where $\bar{\mu}$ is defined in (4.2).

Figure 4.1 shows the convergence behaviour of this approach applied to a randomly-generated linear program with 256 constraints and 1024 variables.

4.1.2 Second approach: conic level set

Renegar's recent work [49] on conic optimization inspires a possible second level-set approach based on interchanging the roles of the affine objective and the conic constraint in (CP). A key step is to define a convex function κ that is nonnegative on the cone $\mathcal{K}$, and positive elsewhere, so that it acts as a surrogate for the conic constraint, i.e.,

$$\kappa(x) \leq 0 \quad \text{if and only if} \quad x \in \mathcal{K}. \tag{4.3}$$

The conic optimization problem can be expressed in entirely functional form as

$$\underset{x}{\text{minimize}} \quad \langle c, x \rangle \quad \text{subject to} \quad \mathcal{A}x = b,\ \kappa(x) \leq 0,$$

which allows us to define the level-set function

$$v(\tau) = \inf_x \{ \kappa(x) \mid \mathcal{A}x = b,\ \langle c, x \rangle \leq \tau \}. \tag{4.4}$$

Renegar gives a procedure for constructing a suitable surrogate function κ under the assumption that $\mathcal{K}$ has a nonempty interior: choose a point $e \in \operatorname{int} \mathcal{K}$ and define $\kappa(x) = -\lambda_{\min}(x)$, where

$$\lambda_{\min}(x) := \inf \{ \lambda \mid x - \lambda e \notin \mathcal{K} \}.$$

In the case of the PSD cone, we may take $e = I$, and then $\lambda_{\min}$ yields the minimum eigenvalue function. As is shown in [49, Prop. 2.1], the function $\lambda_{\min}$ is Lipschitz continuous (with modulus one) and concave, as would be necessary to apply a subgradient method for minimizing κ. The dual of the resulting level-set problem, needed to apply the lower affine-minorant root-finding method, is shown in the second row of Table 4.1, and can be derived using the conjugate of $\lambda_{\min}$; see Lemma A.2.

Renegar derives a novel algorithm along with complexity bounds for CP using the $\lambda_{\min}$ function. A rigorous methodology for applying the level-set scheme, as described in the current paper, requires further research. It is an intriguing research agenda to unify Renegar's explicit complexity bounds with the proposed level-set approach. It is not clear, however, that this approach holds any practical advantage over the least-squares approach described in §4.1.1.

The function $\lambda_{\min}$ is only one example of a surrogate function that satisfies (4.3). Other choices are available for κ that yield suitable value functions (4.4). The best choice ultimately depends on the algorithms that are available for the inexact solution of the corresponding subproblems. For example, we might choose to define the differentiable surrogate function

$$\kappa = \tfrac{1}{2} \operatorname{dist}_{\mathcal{K}}^2, \qquad \text{where} \qquad \operatorname{dist}_{\mathcal{K}}(x) := \inf_{z \in \mathcal{K}} \ \|x - z\|$$

measures the distance to the cone $\mathcal{K}$.

Note the significant differences between the least-squares and conic level-set problems (4.1) and (4.4). For the sake of discussion, suppose that $\mathcal{K}$ is the positive semidefinite cone. The least-squares level-set problem has a smooth objective whose gradient can be easily computed by applying the operator $\mathcal{A}$ and its adjoint, but the constraint set still contains the explicit cone. Projected-gradient methods, for example, require a full eigenvalue decomposition of the steepest-descent step, while the Frank-Wolfe method requires only a single rightmost eigenpair computation. The latter level-set problem, however, can require a potentially more complex procedure to compute a gradient or subgradient, but has an entirely linear constraint set. In this case, projected (sub)gradient methods require a least-squares solve for the projection step.

4.2 Gauge optimization

We now apply the level-set approach to regularized data-fitting problems, restricting the convex functions φ and ρ to be *gauges*—i.e., functions that are additionally nonnegative, positively homogeneous, and vanish at the origin. We assume that the side constraint $x \in \mathcal{X}$ is absent from the formulation $\mathcal{P}_\sigma$. Problems of this type occur in sparsity optimization; basis pursuit (and its "denoising" variant BP_σ) [19] was our very first example in §1. The first two columns of Table 4.2 describe various formulations of current interest, including basis pursuit denoising (BPDN), low-rank matrix recovery [15, 24], a sharp version of the elastic-net problem [62], and gauge optimization [26] in its standard form. The third column shows the level-set problem $\mathcal{Q}_\tau$ needed to evaluate the value function $v(\tau)$, while the fourth column shows the slopes needed to implement the Newton scheme; cf. 2.3.

The dual representation (2.1) can be specialized for this family, and requires some basic facts regarding a gauge function f and its *polar*

$$f^\circ(y) := \inf\left\{\mu > 0 \mid \langle x, y\rangle \le \mu f(x) \text{ for all } x\right\}.$$

When f is a norm, the polar f° is simply the familiar dual norm. There is a close relationship between gauges, their polars, and the support functions of their sublevel sets, as described by the identities [26, Prop. 2.1(iv)]

$$f^\circ = \delta^\star_{[f\le 1]} \quad \text{and} \quad f^\star = \delta_{[f^\circ \le 1]}.$$

We apply these identities to the quantities involving ρ and φ in the expression for the dual representation Φ in (2.1), and deduce

$$\delta^\star_{[\varphi\le\tau]} = \tau\delta^\star_{[\varphi\le 1]} = \tau\varphi^\circ \quad \text{and} \quad \rho^\star = \delta_{[\rho^\circ\le 1]}.$$

Substitute these into Φ to obtain the equivalent expression

$$\Phi(y,\tau) = \langle b, y\rangle - \delta_{[\rho^\circ \le 1]}(-y) - \tau\varphi^\circ(A^* y).$$

We can now write an explicit dual for the level-set problem $\mathcal{Q}_\tau$:

$$\underset{y}{\text{maximize}} \quad \langle b, y\rangle - \tau\varphi^\circ(A^*y) \quad \text{subject to} \quad \rho^\circ(-y) \le 1.$$

In the last three rows of the table, we set $\rho = \|\cdot\|_2$, which is self polar. For BPDN, $\varphi = \|\cdot\|_1$, whose polar is the dual norm $\varphi^\circ = \|\cdot\|_\infty$. For matrix completion, $\varphi = \|\cdot\|_* := \sum_{i=1}^{\min\{m,n\}} \sigma_i(\cdot)$ is the nuclear norm of a n-by-m matrix, which is polar to the spectral norm $\varphi^\circ = \sigma_1(\cdot)$.

Problem	$\mathcal{P}_\sigma$	$\mathcal{Q}_\tau$	$\partial_\tau \Phi(y,\tau)$
gauge optimization	$\min_x \varphi(x)$ s.t. $\rho(Ax-b)\le\sigma$	$\min_x \rho(Ax-b)$ s.t. $\varphi(x)\le\tau$	$-\varphi^\circ(A^*y)$
BPDN	$\min_x \|x\|_1$ s.t. $\|Ax-b\|_2\le\sigma$	$\min_x \|Ax-b\|_2$ s.t. $\|x\|_1\le\tau$	$-\|A^*y\|_\infty$
sharp elast-net	$\min_x \alpha\|x\|_1+\beta\|x\|_2$ s.t. $\|Ax-b\|_2\le\sigma$	$\min_x \|Ax-b\|_2$ s.t. $\alpha\|x\|_1+\beta\|x\|_2\le\tau$	$-\gamma_{\alpha\mathbb{B}_\infty+\beta\mathbb{B}_2}(A^*y)$
matrix completion	$\min_X \|X\|_*$ s.t. $\|\mathcal{A}X-b\|_2\le\sigma$	$\min_x \|\mathcal{A}X-b\|_2$ s.t. $\|X\|_*\le\tau$	$-\sigma_1(\mathcal{A}^*y)$

Table 4.2 Nonsmooth regularized data-fitting.

4.3 Low-rank matrix completion

A range of useful applications that involve missing data can be modeled as matrix completion problems. This modeling approach extends to robust principal-component analysis (RPCA), where we decompose a signal into low-rank and sparse components, and its variants, including its stable version, which allows for noisy measurements. Important examples include applications in recommender systems and system identification [47], alignment of occluded images [45], scene triangulation [60], model selection [18], face recognition, and document indexing [14].

These problems can be formulated generally as

$$\underset{X\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad \varphi(X) \quad \text{subject to} \quad \rho(\mathcal{A}X-b)\le\sigma, \tag{4.5}$$

where b is a vector of observations, the linear operator $\mathcal{A}$ encodes information about the measurement process, the objective φ encourages the low-rank and possibly other structure in the solution, and the constraint ρ measures the misfit between $\mathcal{A}X$ and b. If we wish to require $\mathcal{A}X=b$, we can simply set $\sigma=0$ and choose any nonnegative convex function ρ that vanishes only at the origin, e.g., the 2-norm.

We categorize the family of low-rank problems into *symmetric* and *asymmetric* classes. For each case, we describe a basic formulation and how the level-set approach leads to implementable algorithms with computational kernels that scale well with problem size.

4.3.1 Symmetric problems.

The symmetric class of problems aims to recover a low-rank PSD matrix, with a linear operator $\mathcal{A}$ that maps between the space of symmetric $n\times n$ matrices and m-vectors. We define the objective of (4.5) by

$$\varphi_1(X)=\operatorname{tr}(X)+\delta_{\mathcal{S}^n_+}(X). \tag{4.6}$$

Problem (4.5) then reduces to finding a PSD matrix with minimum trace that satisfies the constraints $\rho(\mathcal{A}X-b)\le\sigma$. It is straightforward to extend this formulation to optimization over Hermitian matrices, so that it includes important applications such as phase retrieval, which aims to recover phase information about a signal (e.g., and image) from a series of magnitute-only measurements [17, 36, 54]. For simplicity, we focus here only on real-valued matrices.

4.3.2 Asymmetric problems.

The asymmetric class of matrix-recovery problems does not require definiteness of X. In this case, the linear operator $\mathcal{A}$ on $\mathbb{R}^{m\times n}$ is not restricted to symmetric matrices. We define the objective of (4.5) by

$$\varphi_2(X) = ||X||_* \tag{4.7}$$

This formulation captures matrix completion [46], bi-convex compressed sensing [35], and robust PCA [16, 61].

Example 4.1 (Robust PCA) We give an example of how to a problem that is not in the form of (4.5) can be recast to fit into the required formulation. The stable version of the RPCA problem [57] aims to decompose an m-by-n matrix B as a sum of a low-rank matrix and a sparse matrix via the problem

$$\operatorname*{minimize}_{L,S} \quad \lambda_L\|L\|_* + \lambda_S\|S\|_1 + \tfrac{1}{2}\|\mathcal{A}[L-B]-S\|_F^2. \tag{4.8}$$

Here the operator $\mathcal{A}$ is often a mask for the known elements of B. The goal is to obtain a low-rank approximation to Y where the deviation from the known elements of B are sparse. The positive parameters λ_L and λ_S balance the rank of L against the sparsity of the residual S, and the least-squared misfit.

We show how this model might be recast within the formulation (4.5). The first step is based on the observation that, as a function of S, the objective is the Moreau envelope of the 1-norm evaluated at $\mathcal{A}(L-B)$, or equivalently, the Huber function on $\mathcal{A}(L-B)$. In particular,

$$\inf_S \left\{\lambda_S\|S\|_1 + \tfrac{1}{2}\|\mathcal{A}[L-B]-S\|_F^2\right\} = \rho_{\lambda_S}(\mathcal{A}[L-B]),$$

where

$$\rho_\alpha(R) = \sum_{i,j} \begin{cases} \frac{1}{2}r_{ij}^2 & \text{if } |r_{ij}| \le \alpha, \\ \alpha(|r_{ij}| - \frac{1}{2}\alpha) & \text{otherwise,} \end{cases}$$

is the Huber function. For some nonnegative parameter σ, we can then reinterpret (4.8) as the problem of finding the lowest-rank approximation to B subject to a bound on a robust measure of misfit. This yields the related problem

$$\operatorname*{minimize}_{L} \quad ||L||_* \quad \text{subject to} \quad \rho_{\lambda_L}(\mathcal{A}[L-B]) \le \sigma.$$

In some contexts, this formulation may be preferable to (4.8) if there is a target level of misfit as measured by the constraint.

4.3.3 Level-set approach and the Frank-Wolfe oracle

We apply the level-set approach to (4.5) and exchange the roles of the regularizing function φ and the misfit $\rho(\mathcal{A}X - b)$. The objective function φ_1 for the symmetric case vanishes at the origin, and is convex and positively homogeneous; it is thus a gauge. The second objective function φ_2 is simply a norm. For both cases, we can use the first row of Table 4.2 to determine the corresponding level-set subproblem and affine minorants based on dual certificates. The corresponding level-set subproblem $\mathcal{Q}_\tau$, which defines the value function, is

$$v(\tau) := \min_X \left\{ \rho(\mathcal{A}X - b) \mid \varphi(X) \le \tau \right\}.$$

We use the polar calculus described by Friedlander et al. [26, §7.2.1] and the definition of the dual norm to obtain the required polar functions

$$\varphi_1^\circ(Y) = \max\{0,\ \lambda_1(Y)\} \qquad \text{and} \qquad \varphi_2^\circ(Y) = \sigma_1(Y)$$

for the symmetric (4.6) and asymmetric (4.7) cases, respectively.

The evaluation of the affine minorant oracle requires an approximate solution of the optimization problem that defines the value function v, and computation of either an extreme eigenvalue or singular value to determine an affine minorant. The Frank-Wolfe algorithm [25, 29] is well suited for evaluating the required quantities, Each iteration of Frank-Wolfe updates the solution estimate via $X^+ \leftarrow X + \alpha(\widehat{X} - X)$, where α is a step length, $\widehat{X}$ is a solution of the linearized subproblem

$$\underset{\widehat{X}}{\text{maximize}}\ \langle G, \widehat{X} \rangle \ \ \text{subject to}\ \ \varphi(\widehat{X}) \le \tau, \tag{4.9}$$

and $G := \mathcal{A}^*\nabla\rho(\mathcal{A}X - b)$ is the gradient of the constraint function evaluated at the current primal iterate X. Note that the steplength in this case is easily obtained as the minimizer of the quadratic objective along the intersection of $[\varphi \le \tau]$ and the ray $X + \mathbb{R}_+(\widehat{X} - X)$.

Solutions of the subproblem (4.9) depend on the extreme eigenvalues or singular values of G [29, §4.2]. For the symmetric case (4.6), the constraint

$$\varphi_1(\widehat{X}) = \operatorname{tr}(\widehat{X}) + \delta_{\mathcal{S}^n_+}(\widehat{X}) \le \tau \quad \text{is equivalent to} \quad \operatorname{tr}(\widehat{X}) \le \tau,\ \widehat{X} \succeq 0.$$

For the asymmetric case, the constraint $\varphi_2(\widehat{X}) \le \tau$ is simply $\|\widehat{X}\|_* \le \tau$. The solutions $\widehat{X}_1$ and $\widehat{X}_2$, respectively, of the subproblems (4.9) corresponding to the symmetric and asymmetric cases, have the form

$$\begin{aligned} \widehat{X}_1 &= U\mathrm{Diag}\,(\xi_i)U^T, \\ \widehat{X}_2 &= U\mathrm{Diag}\,(\xi_i)V^T, \end{aligned} \qquad \text{with} \qquad \sum_{i=1}^k \xi_i = \tau,\ \xi_i \ge 0.$$

For the symmetric case, the orthonormal n-by-k matrix U collects the k eigenvectors of G corresponding to $\lambda_1(G)$. For the asymmetric case, the m-by-k matrix U and n-by-k matrix V, respectively, collect the k left- and right-singular vectors of G corresponding to the leading singular value $\sigma_1(G)$. In both cases, Krylov-based eigensolvers, such as ARPACK [31] can be used for the required eigenvalue and singular-value computation. If matrix-vector products with the matrix $\mathcal{A}^*y$ and its adjoint are computationally inexpensive, the computation of a few rightmost eigenvalue/eigenvector pairs (resp., maximum singular value/vector pairs) is much cheaper than the computation of the entire spectrum, as required by a method based on projections onto the feasible region. Such circumstances are common, for example when the operator $\mathcal{A}$ is sparse or it is accessible through a fast Fourier transform.

The next example illustrates an application where the operator $\mathcal{A}$ is accessibly only via its action on a matrix. It also gives us an opportunity to describe how the level-set method can be easily adapted to to solve a *maximization* problem, which requires computing the right-most root to the corresponding value function.

Example 4.2 (Euclidean distance completion) A common problem in distance geometry is the inverse problem: given only local pairwise Euclidean distance measurements among a set of points, recover their location in space. Formally, given a weighted undirected graph $G = (V, E, \omega)$ with a vertex set $V = \{1, \ldots, n\}$, and a target dimension r, the Euclidean distance completion problem asks to determine a collection of points $p_1, \ldots, p_n$ in $\mathbb{R}^r$ approximately satisfying

$$\|p_i - p_j\|^2 = \omega_{ij} \qquad \text{for all edges} \qquad ij \in E.$$

This problem is also often called ℓ_2 graph embedding and appears in wireless networks, statistics, robotics, protein reconstruction, and manifold learning [34].

A popular convex relaxation for this problem was introduced by Weinberger et al. [55] and extensively studied by a number of authors [8, 9, 22]:

$$\underset{X}{\text{maximize}} \;\; \operatorname{tr}(X) \quad \text{subject to} \quad \begin{aligned} &\|\mathcal{P}_E \circ \mathcal{K}(X) - \omega\| \le \sigma, \\ &Xe = 0, \;\; X \succeq 0, \end{aligned} \tag{4.10}$$

where $\mathcal{K}\colon \mathcal{S}^n \to \mathcal{S}^n$ is the mapping $[\mathcal{K}(X)]_{ij} = X_{ii} + X_{jj} - 2X_{ij}$ and $\mathcal{P}_E(\cdot)$ is the canonical projection of a matrix onto entries indexed by the edge set E. Indeed, if X is a rank r feasible matrix, we may factor it into $X = PP^T$, where P is an $n \times r$ matrix; the rows of P are the points $p_1, \ldots, p_n \in \mathbb{R}^r$ we seek. The constraint $Xe = 0$ ensures that the points p_i are centered around the origin. Note that this formulation *maximizes* the trace $\operatorname{tr}(X) = \frac{1}{2n}\sum_{i,j=1}^n \|p_i - p_j\|^2$, which helps to "flatten" the realization of the graph.

It is well-known that for $\sigma = 0$, the constraints of problem (4.10) do not admit a strictly feasible solution [21, 22, 30]. In particular, for small positive σ, the feasible region is very thin and the solution to the problem is unstable. As a result, algorithms maintaining feasibility are likely to have difficulties. In contrast, the level-set approach is an infeasible method, and hence the poor conditioning of the underlying problem does not play a major role.

The least-squares level-set problem that corresponds to the minimization formulation of (4.10) is

$$\begin{aligned} &\text{minimize} && \|\mathcal{P}_E \circ \mathcal{K}(X) - \omega\| \\ &\text{subject to} && \operatorname{tr}(X) \ge \tau, \;\; Xe = 0, \;\; X \succeq 0. \end{aligned} \tag{4.11}$$

The inequality $\operatorname{tr}(X) \ge \tau$ takes into account that the original formulation (4.10) is a *maximization* problem. As a result, the root-finding method on the value function corresponding to (4.11) approaches the optimal value $\tau_* = \text{OPT}$ from the right. To initialize the approximate Newton scheme, we need an upper bound τ_0 on the objective function, which is easily available from the diameter of the graph.

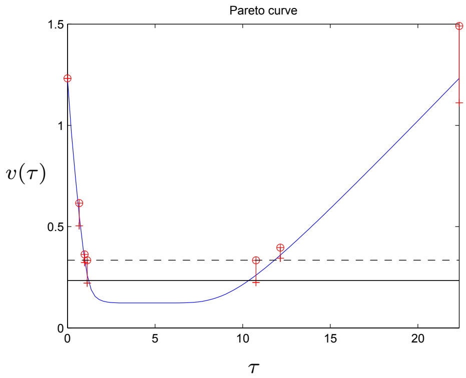


Fig. 4.2 A plot of the value function $v(\tau)$ of (4.11). Newton's method converges to either the min- or max-trace solution, depending on initialization. To solve (4.10), we need the maximal root. Here, $\sigma = 0.25$, indicated by the solid horizontal line.

The gradient of the objective function is as sparse as the edge set E, and the linear subproblem over the feasible region requires computing only a maximal eigenvalue on a sparse matrix [22]. This makes the problem (4.11) ideally suited for the Frank-Wolfe algorithm. The dual problem of (4.11) takes the form

$$\underset{y\in\mathbb{R}^E,\ \|y\|_2\le 1}{\text{maximize}} \quad \Phi(y,\tau) := \langle y, \omega\rangle - 2\tau\lambda_1^{e^\perp}(\operatorname{Diag}(Ye) - Y),$$

where $\lambda_1^{e^\perp}(A)$ is the maximal eigenvalue of the restriction of the matrix A to the space orthogonal to e, and $Y = \mathcal{P}_E^*(y)$ is diagonal matrix formed from the vector y padded with zeros. The expression of the dual objective follows from the fact that $\mathcal{K}^*\mathcal{P}_E^*(y) = 2(\operatorname{Diag}(Ye) - Y)$. As described in §2.4, we use the sequence $\{y_i\}$ of dual certificates generated by the Frank-Wolfe algorithm to establish at iteration k an affine minorant at defined by

$$\ell_k = \min_{i=1,\dots,k} \Phi(y_i,\tau) \quad \text{and} \quad s_k = 2\lambda_1^{e^\perp}(\operatorname{Diag}(Y_k e) - Y_k)$$

where $Y_k = \mathcal{P}_E^*(y_k)$. A full derivation and extensive numerical results are given [22]. Figure 4.2 shows the iterations of Newton's method applied to this problem.

A Proofs

Proof (Proof of Theorem 2.1) Since f is convex, the subdifferential $\partial f(\tau)$ is nonempty for all $\tau \in (a,b)$. The claim concerning finite termination is easy to deduce from convexity; we leave the details to the reader. Suppose neither sequence terminates finitely at τ_*. Let us first consider the Newton iteration. Convexity of f immediately implies that the sequence τ_i is well-defined and satisfies $\tau_0 < \tau_1 < \tau_2 < \cdots < \tau_*$. Monotonicity of the subdifferential then implies $g_0 \le g_1 \le g_2 \le \cdots \le g_* < 0$. Due to the inequalities $f(\tau_*) + \bar{g}(\tau_k - \tau_*) \le f(\tau_k)$ and $g_k < 0$, we have

$$\frac{f(\tau_k) - f(\tau_*)}{g_k} \le -\frac{g_*}{g_k}(\tau_* - \tau_k),$$

and so

$$0 < \tau_* - \tau_{k+1} = \tau_* - \tau_k + \frac{f(\tau_k) - f(\tau_*)}{g_k} \le \left(1 - \frac{g_*}{g_k}\right)(\tau_* - \tau_k).$$

Upper semi-continuity of ∂f on its domain implies $g_k \uparrow g_*$. Hence τ_k converge q-superlinearly to τ_*.

Now consider the secant iteration. As in the Newton iteration, it is immediate from convexity that the sequence τ_i is well-defined and satisfies $\tau_0 < \tau_1 < \tau_2 < \cdots < \tau_*$. Monotonicity of the subdifferential then implies $g_0 \le g_1 \le g_2 \le \cdots \le g_* < 0$. We have

$$0 < g_*(\tau_k - \tau_*) \le f(\tau_k) - f(\tau_*),$$

and $f(\tau_{k-1}) + g_{k-1}(\tau_k - \tau_{k-1}) \le f(\tau_k)$, and hence

$$\frac{\tau_k - \tau_{k-1}}{f(\tau_k) - f(\tau_{k-1})}(f(\tau_k) - f(\tau_*)) \le \frac{f(\tau_k) - f(\tau_*)}{g_{k-1}} < 0.$$

Combining the two inequalities yields

$$\frac{f(\tau_k) - f(\tau_*)}{f(\tau_k) - f(\tau_{k-1})}(\tau_k - \tau_{k-1}) \le \frac{f(\tau_k) - f(\tau_*)}{g_{k-1}} \le \frac{g_*}{g_{k-1}}(\tau_k - \tau_*) < 0.$$

Consequently, we deduce

$$0 < \tau_* - \tau_{k+1} = \tau_* - \tau_k + \frac{f(\tau_k) - f(\tau_*)}{f(\tau_k) - f(\tau_{k-1})}(\tau_k - \tau_{k-1}) \le \left(1 - \frac{g_*}{g_{k-1}}\right)(\tau_* - \tau_k).$$

The result follows.

Proof (Proof of Theorem 2.2) It is easy to see by convexity that the iterates τ_k are strictly increasing and satisfy $f(\tau_k) > 0$. For each index $j \ge 2$ for which the algorithm has not terminated, define the following quantities:

$$h_j := \tau_j - \tau_{j-1}, \qquad \theta_j := \frac{s_j}{s_{j-1}}, \qquad \text{and} \qquad \gamma_j := \frac{\ell_j}{\ell_{j-1}}.$$

Note that using the equation $\tau_{j-1} - \tau_j = \frac{\ell_{j-1}}{s_{j-1}}$, we can write $\theta_j = \frac{u_{j-1} - \ell_j}{\ell_{j-1}}$. Clearly then the bound, $0 \le \theta_j \le \alpha - \gamma_j$, is valid. Define now constants $\beta_j \in [0,1]$ by the equation $\gamma_j = \beta_j \alpha$. Suppose $k \ge 2$ is an index at which the algorithm has not terminated, i.e., $u_k > \epsilon$. Taking into account the inequality $\ell_k \ge \frac{u_k}{\alpha} > \frac{\epsilon}{\alpha}$, we deduce

$$\frac{\epsilon}{\alpha} \le \ell_k = \ell_1 \prod_{j=2}^{k} \gamma_j \le C\alpha^{k-1} \prod_{j=2}^{k} \beta_j. \tag{A.1}$$

The defining equation for τ_{k+1} and the definition of θ_j yield the equality

$$h_{k+1} = \frac{\ell_k}{|s_k|} = \frac{\ell_k}{|s_1|} \cdot \prod_{j=2}^{k} \theta_j^{-1}.$$

The bounds $\tau_* - \tau_1 \ge h_{k+1}$, $\ell_k \ge \frac{\epsilon}{\alpha}$, and $\theta_j \le \alpha - \gamma_j$ imply

$$\tau_* - \tau_1 \ge \frac{\ell_k}{|s_1|} \cdot \prod_{j=2}^{k} \theta_j^{-1} \ge \frac{\epsilon}{\alpha|s_1|}(\alpha^{-1})^{k-1} \prod_{j=2}^{k} (1 - \beta_j)^{-1},$$

and rearranging gives

$$\epsilon \le (\tau_* - \tau_1)|s_1|\alpha^k \prod_{j=2}^{k} (1 - \beta_j) \le C\alpha^k \prod_{j=2}^{k} (1 - \beta_j). \tag{A.2}$$

Combining (A.1) and (A.2), we get

$$\epsilon \le C\alpha^k \min\left\{ \prod_{j=2}^{k} \beta_j,\ \prod_{j=2}^{k} (1 - \beta_j) \right\}. \tag{A.3}$$

One the other hand, observe

$$\left(\prod_{j=2}^{k}\beta_j\right)\left(\prod_{j=2}^{k}(1-\beta_j)\right)=\prod_{j=2}^{k}\beta_j(1-\beta_j)\leq 0.5^{2(k-1)},$$

and hence

$$\min\left\{\prod_{j=2}^{k}\beta_j,\ \prod_{j=2}^{k}(1-\beta_j)\right\}\leq 0.5^{k-1}. \tag{A.4}$$

Combining equations (A.4) and (A.3), the claimed estimate $k-1\leq\log_{2/\alpha}\left(\frac{\alpha C}{\epsilon}\right)$ follows.

Proof (Proof of Theorem 2.3) The proof is identical to the proof of Theorem 2.2, except for some minor modifications. The only nontrivial change is how we arrive at the bound $\theta_j\leq\alpha-\gamma_j$. For this, observe $\tau_{j-1}-\tau_j=\ell_{j-1}/s_{j-1}$, and because the function $\tau\mapsto\ell_j+s_j(\tau-\tau_j)$ minorizes f, we see

$$u_{j-1}\geq\ell_j+s_j(\tau_{j-1}-\tau_j)=\ell_j+s_j\left(\frac{\ell_{j-1}}{s_{j-1}}\right)=\ell_j+\theta_j\ell_{j-1}.$$

After rearranging, we get the desired upper bound on θ_j:

$$\theta_j\leq\frac{u_{j-1}-\ell_j}{\ell_{j-1}}\leq\alpha-\gamma_j.$$

Finally, we remark that with the approximate Newton method, we can start indexing at $j=0$ instead of $j=1$. This explains the different constants in the convergence result.

Lemma A.1 (Concavity of the parametric support function) *For any convex function $f\colon\mathbb{R}^n\to\overline{\mathbb{R}}$ and vector $z\in\mathbb{R}^n$, the univariate function $t\mapsto\delta^*_{[f\leq t]}(z)$ is concave.*

Proof It follows from convexity of f that

$$\lambda\cdot[f\leq a]+(1-\lambda)\cdot[f\leq b]\subseteq[f\leq\lambda a+(1-\lambda)b]\qquad\forall a,b\in\mathbb{R}\text{ and }\lambda\in[0,1],$$

where $[f\leq\alpha]$ defines the α-level set of f, and the summation of the level sets indicates their Minkowski (i.e., direct) sum. Moreover, for any convex sets $\mathcal{C}$ and $\mathcal{D}$ such that $\mathcal{C}\subseteq\mathcal{D}$, $\delta^*_{\mathcal{C}}\leq\delta^*_{\mathcal{D}}$. Thus,

$$\lambda\cdot\delta^*_{[f\leq a]}(z)+(1-\lambda)\cdot\delta^*_{[f\leq b]}(z)=\delta^*_{\lambda\cdot[f\leq a]+(1-\lambda)\cdot[f\leq b]}(z)\leq\delta^*_{[f\leq\lambda a+(1-\lambda)b]}(z),$$

which implies concavity of the function at hand..

Proof (Proof of Proposition 3.1) For this proof only, let $\|\cdot\|$ denote the 2-norm. Note the inclusion $s/\|y\|\in\partial_\tau\Phi_1\left(y/\|y\|,\tau\right)$. Use the same computation from (2.2) to deduce that the affine function

$$\tau'\mapsto(\hat{\ell}-\sigma)-\frac{s}{\|y\|}(\tau'-\tau)$$

minorizes f_1.

From the definition of $\hat{\ell}$, Φ_1, and Φ_2, it follows that

$$\frac{u-\sigma}{\hat{\ell}-\sigma}=\frac{(u-\sigma)\|y\|}{\Phi_2(y,\tau)+\frac{1}{2}\|y\|^2-\sigma\|y\|}=\frac{2(u-\sigma)\|y\|}{\ell^2+\|y\|^2-2\sigma\|y\|}. \tag{A.5}$$

Taking into account the equivalence

$$\frac{u-\sigma}{\ell-\sigma}\leq\alpha\quad\Longleftrightarrow\quad\frac{u+(\alpha-1)\sigma}{\alpha}\leq\ell,$$

we deduce

$$\ell^2 + \|y\|^2 - 2\sigma\|y\| \geq \alpha^{-2}\Big((u + (\alpha - 1)\sigma)^2 + \|\alpha y\|^2 - 2\sigma\alpha\|\alpha y\|\Big) \geq 2\alpha^{-1}(u - \sigma)\|y\|,$$

where the rightmost inequality follows from the computation

$$\begin{aligned}(u + [\alpha - 1]\sigma)^2 &+ \|\alpha y\|^2 - 2\alpha\sigma\|\alpha y\| - 2(u - \sigma)\|\alpha y\| \\ &= (u + [\alpha - 1]\sigma)^2 + \|\alpha y\|^2 - 2\|\alpha y\|(u + [\alpha - 1]\sigma) \\ &= (u + [\alpha - 1]\sigma - \|\alpha y\|)^2 \geq 0.\end{aligned}$$

Because the right-hand side of (A.5) is non-negative, we can deduce that $\hat{\ell} \geq \sigma$. Finally, the required inequality $(u - \sigma)/(\hat{\ell} - \sigma) \leq \alpha$ also follows from (A.5).

Lemma A.2 $(-\lambda_{\min})^\star(y) = \delta_{\mathcal{S}}(-y)$*, where* $\mathcal{S} = \mathcal{K}^* \cap \{x \mid \langle e, x\rangle = 1\}$*.*

Proof The following formula is established in [49]:

$$\partial(-\lambda_{\min})(x) = \{-y \mid \langle y, e\rangle = 1,\ \langle y, z - (x - \lambda_{\min}(x)e)\rangle \geq 0 \text{ for all } z \in \mathcal{K}\}$$

or equivalently

$$\begin{aligned}\partial(-\lambda_{\min})(x) &= \{-y \mid \langle y, e\rangle = 1, -y \in N_{\mathcal{K}}\,(x - \lambda_{\min}(x)e)\,\} \\ &= \big\{-y \mid \langle y, e\rangle = 1,\ y \in \mathcal{K}^*,\ 0 = \lambda_{\min}(x) - \langle y, x\rangle\,\big\}\,.\end{aligned}$$

Here the symbol $N_{\mathcal{K}}$ denotes the normal cone to $\mathcal{K}$. Now for any $y \in \partial(-\lambda_{\min})(x)$, we have $\langle x, y\rangle = -\lambda_{\min}(x)$. Observe range $\partial(-\lambda_{\min}) = -\mathcal{S}$. Hence by the equality in the Fenchel-Young inequality, for any $y \in -\mathcal{S}$, we have $(-\lambda_{\min})^\star(y) = 0$. On the other hand, for any y with $\langle y, e\rangle \neq -1$, we have $(-\lambda_{\min})^\star(y) \geq \langle te, y\rangle - (-\lambda_{\min})(te) = t(\langle y, e\rangle + 1)$ for any $t \geq 0$. Letting $t \to \infty$, we deduce $(-\lambda_{\min})^\star(y) = +\infty$. Similarly, consider $y \notin -\mathcal{K}^*$. Then we may find some $x \in \mathcal{K}$ satisfying $\langle x, y\rangle > 0$. We deduce $(-\lambda_{\min})^\star(y) \geq \langle tx, y\rangle - (-\lambda_{\min})(tx) = t(\langle y, x\rangle - (-\lambda_{\min})(x))$ for any $t \geq 0$. Letting $t \to \infty$, we deduce $(-\lambda_{\min})^\star(y) = +\infty$. We deduce that $(-\lambda_{\min})^\star$ is the indicator function of $-\mathcal{S}$, as claimed.

Remark A.1 (Projection onto a conic slice sets) This remark is standard. Fix a proper convex cone $\mathcal{K}$ and consider the projection problem

$$\min_x\ \left\{\tfrac{1}{2}\|x - z\|^2 \mid \langle c, x\rangle = 1,\ x \in \mathcal{K}\right\}.$$

Equivalently, we can consider the univariate concave maximization problem

$$\begin{aligned}\max_\beta \min_{x\in\mathcal{K}} L(x, \beta) &= \max_\beta \min_{x\in\mathcal{K}} \tfrac{1}{2}\|x - z\|^2 + \beta(\langle c, x\rangle - 1) \\ &= \max_\beta \min_{x\in\mathcal{K}} \tfrac{1}{2}\|x - (z - \beta c)\|^2 + \beta(\langle c, z\rangle - 1) - \tfrac{1}{2}\beta^2\|c\|^2 \\ &= \max_\beta\ \tfrac{1}{2}\mathrm{dist}^2_{\mathcal{K}}(z - \beta c) + \beta(\langle c, z\rangle - 1) - \tfrac{1}{2}\beta^2\|c\|^2.\end{aligned}$$

We can solve this problem for example by bisection, provided projections onto $\mathcal{K}$ are available.

Acknowledgements The authors extend their sincere thanks to three anonymous referees who provided an extensive list of corrections and suggestions that helped us to streamline our presentation.

References

1. Aravkin, A. Y., J. Burke, and M. P. Friedlander. 2013. Variational properties of value functions. *SIAM J. Optimization* 23 (3): 1689–1717.
2. Bach, F. 2015. Duality between subgradient and conditional gradient methods. *SIAM J. Optim.* 25 (1): 115–129.
3. Beck, A., and M. Teboulle. 2009. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences* 2 (1): 183–202.
4. Beck, Amir, and Marc Teboulle. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* 31 (3): 167–175.
5. Ben-Tal, A., and A. Nemirovski. 2001. *Lectures on modern convex optimization. Mps/siam series on optimization.*
6. Bertsekas, Dimitri P. 2015. *Convex optimization algorithms.* Massachusetts: Athena Scientific.
7. Biswas, P., and Y. Ye. 2004. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on information processing in sensor networks*, 46–54. ACM. ACM.
8. Biswas, P., and Y. Ye. 2006. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale optimization methods and applications.* Vol. 82 of *Nonconvex optim. appl.*, 69–84. Springer.
9. Biswas, P., T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang. 2006a. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Trans. Autom. Sci. Eng.* 3 (4): 360–371.
10. Biswas, P., T. C. Lian, T. C. Wang, and Y. Ye. 2006b. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)* 2 (2): 188–220.
11. Borwein, J. M., and A. S. Lewis. 2000. *Convex analysis and nonlinear optimization. Cms books in mathematics/ouvrages de mathématiques de la smc, 3.* New York: Springer. Theory and examples.
12. Boyd, S., N. Parikh, R. Chu, B. Peleato, and J. Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3 (1): 1–122.
13. Brucker, P. 1984. An $o(n)$ algorithm for quadratic knapsack problems. *Oper. Res. Lett.* 3 (3): 163–166.
14. Candès, E. J., X. Li, Y. Ma, and J. Wright. 2011. Robust principal component analysis? *J. Assoc. Comput. Mach.* 58 (3): 1–37.
15. Candès, E. J., and T. Tao. 2010. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Info. Th.* 56 (5): 2053–2080.
16. Candès, E. J., X. Li, Y. Ma, and J. Wright. 2011. Robust principal component analysis? *J. ACM* 58 (3): 11–11137.
17. Candès, Emmanuel J, Thomas Strohmer, and Vladislav Voroninski. 2012. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Commun. Pur. Appl. Ana.*.
18. Chandrasekaran, V., P. A. Parrilo, and A. S. Willsky. 2012. Latent variable graphical model selection via convex optimization. *Ann. Stat.* 40 (4): 1935–2357.
19. Chen, S. S., D. L. Donoho, and M. A. Saunders. 1999. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.* 20 (1): 33–61.
20. Cox, Bruce, Anatoli Juditsky, and Arkadi Nemirovski. 2014. Dual subgradient algorithms for large-scale nonsmooth learning problems. *Mathematical Programming* 148 (1-2): 143–180.
21. Drusvyatskiy, D., G. Pataki, and H. Wolkowicz. 2015. Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.* 25 (2): 1160–1178.
22. Drusvyatskiy, D., N. Krislock, Y. L. Voronin, and H. Wolkowicz. 2014. Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. *Preprint, arXiv:1410.6852.*
23. Ennis, R. h., and G. C. McGuire. 2001. *Computer algebra recipes: A gourmet's guide to the mathematical models of science.* Springer.
24. Fazel, M. 2002. Matrix rank minimization with applications. PhD diss, Elec. Eng. Dept, Stanford University.
25. Frank, M., and P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Res. Logist. Quart.* 3: 95–110.

26. Friedlander, M. P., I. Macêdo, and T. K. Pong. 2014. Gauge optimization and duality. *SIAM J. Optim.* 24 (4): 1999–2022. doi:10.1137/130940785.
27. Gabay, D., and B. Mercier. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications* 2 (1): 17–40.
28. Harchaoui, Z., A. Juditsky, and A. Nemirovski. 2015. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Math. Program.* 152 (1-2, Ser. A): 75–112.
29. Jaggi, Martin. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proc. 30th intern. conf. machine learning (icml-13)*, 427–435.
30. Krislock, N., and H. Wolkowicz. 2010. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM J. Optim.* 20 (5): 2679–2708.
31. Lehoucq, Richard B, Danny C Sorensen, and Chao Yang. 1998. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, Vol. 6. SIAM.
32. Lemaréchal, C. 1975. An extension of Davidon methods to nondifferentiable problems. *Math. Programming Stud.* 3: 95–109.
33. Lemaréchal, C, A. Nemirovskii, and Y. Nesterov. 1995. New variants of bundle methods. *Math. Programming* 69 (1, Ser. B): 111–147. Nondifferentiable and large-scale optimization (Geneva, 1992).
34. Liberti, L., C. Lavor, N. Maculan, and A. Mucherino. 2014. Euclidean distance geometry and applications. *SIAM Review* 56 (1): 3–69.
35. Ling, Shuyang, and Thomas Strohmer. 2015. Self-calibration and biconvex compressive sensing. *CoRR* abs/1501.06864. `http://arxiv.org/abs/1501.06864`.
36. Luke, Russell, James Burke, and Richard Lyons. 2002. Optical wavefront reconstruction: theory and numerical methods. *SIAM Review* 44: 169–224.
37. Markowitz, H. M. 1987. *Mean-variance analysis in portfolio choice and capital markets.* Frank J. Fabozzi Associates, New Hope, Pennsylvania.
38. Marquardt, D. 1963. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics* 11: 431–441.
39. Miettinen, K. 1999. *Nonlinear multi-objective optimization.* Springer.
40. Morrison, D. D. 1960. Methods for nonlinear least squares problems and convergence proofs. In *Proceedings of the seminar on tracking programs and orbit determination*, eds. J. Lorell and F. Yagi, 1–9. Pasadena, USA: Jet Propulsion Laboratory.
41. Nemirovski, A. 2004. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.* 15 (1): 229–251.
42. Nesterov, Y. 2004. *Introductory lectures on convex optimization.* Dordrecht, The Netherlands: Kluwer Academic.
43. Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Math. Program.* 103 (1): 127–152.
44. Osborne, Michael R, Brett Presnell, and Berwin A Turlach. 2000. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis* 20 (3): 389–403.
45. Peng, Y., A. Ganesh, J. Wright, W. Xu, and Y. Ma. 2012. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 34 (11): 2233–2246.
46. Recht, B., M. Fazel, and P. A. Parrilo. 2010a. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review* 52 (3): 471–501.
47. Recht, Benjamin, Maryam Fazel, and Pablo A. Parrilo. 2010b. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* 52 (3): 471–501.
48. Renegar, J. 1995. Linear programming, complexity theory and elementary functional analysis. *Math. Programming* 70 (3, Ser. A): 279–351.
49. Renegar, J. 2015. A framework for applying subgradient methods to conic optimization problems. *Preprint arXiv:1503.02611.*
50. Rockafellar, R T. 1970. *Convex Analysis. Priceton landmarks in mathematics.* Princeton University Press.
51. Tseng, P. 2010. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Prog.* 125 (2): 263–295.
52. van den Berg, E., and M. P. Friedlander. 2011. Sparse optimization with least-squares

constraints. *SIAM J. Optimization* 21 (4): 1201–1229.

53. van den Berg, E., and M. P. Friedlander. 2008. Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comp.* 31 (2): 890–912.
54. Waldspurger, Irène, Alexandre d'Aspremont, and Stéphane Mallat. 2015. Phase recovery, maxcut and complex semidefinite programming. *Math. Prog.* 149 (1-2): 47–81.
55. Weinberger, Kilian Q., Fei Sha, and Lawrence K. Saul. 2004. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on machine learning. Icml '04*, 106. New York, NY, USA: ACM.
56. Wolfe, P. 1975. A method of conjugate subgradients for minimizing nondifferentiable functions. *Math. Programming Stud.* 3: 145–173.
57. Wright, J., A. Ganesh, S. Rao, and Y. Ma. 2009a. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *Neural information processing systems (nips)*.
58. Wright, Stephen J, Robert D Nowak, and Mário AT Figueiredo. 2009b. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing* 57 (7): 2479–2493.
59. Yin, Wotao. 2010. Analysis and generalizations of the linearized bregman method. *SIAM J. Imag. Sci.* 3 (4): 856–877.
60. Zhang, Z., X. Liang, A. Ganesh, and Y. Ma. 2011. TILT: Transform icpw12nvariant low-rank textures. In *Computer vision – accv 2010*, eds. R. Kimmel, R. Klette, and A. Sugimoto. Vol. 6494 of *Lecture notes in computer science*, 314–328. Springer.
61. Zheng, Peng, Aleksandr Y. Aravkin, Karthikeyan Natesan Ramamurthy, and Jayaraman Jayaraman Thiagarajan. 2017. Learning robust representations for computer vision. In *The ieee international conference on computer vision (iccv) workshops*.
62. Zou, H., and T. Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* 67: 301–320.