Algorithms for convex optimization with applications to data science

Scott Roy

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

Dmitriy Drusvyatskiy, Chair

James V. Burke

Aleksandr Aravkin

Program Authorized to Offer Degree:
Mathematics

University of Washington

Abstract

Algorithms for convex optimization
with applications to data science

Scott Roy

Chair of the Supervisory Committee:
Assistant Professor Dmitriy Drusvyatskiy
Department of Mathematics

Convex optimization is more popular than ever, with extensive applications in statistics, machine learning, and engineering. Nesterov introduced optimal first-order methods for large scale convex optimization in the 1980s, and extremely fast interior point methods for small-to-medium scale convex optimization emerged in the 1990s. Today there is little reason to prefer modelling with linear programming over convex programming for computational reasons. Nonetheless, there is room to improve the already sophisticated algorithms for convex optimization.

The thesis makes three primary contributions to convex optimization. First, the thesis develops new, near optimal barriers for generalized power cones. This is relevant because the performance of interior point methods depends on representing convex sets with "small parameter" barriers. Second, the thesis introduces an intuitive, first-order method that achieves the best theoretical convergence rate and has better performance in practice than Nesterov's method. The thesis concludes with a framework for reformulating a convex program by interchanging the objective function and a constraint function. The approach is illustrated on several examples.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I'd like to thank my collaborators, Sasha Aravkin, Jim Burke, Dima Drusvyatskiy, Maryam Fazel, Michael Friedlander, and Lin Xiao. Working with so many great mathematicians, computer scientists, and engineers is the best education anyone can have. I especially want to thank my advisor, Dima. Dima is smart, enthusiastic, and cheerful. I am much better today for working with him.

I also want to thank all the people I love and who make life so great. In particular, I want to thank my parents, Rick and Jeanne, for sacrificing so much for me, my brothers Sam and Michael, my uncle Dan for welcoming me when I moved to Seattle, my officemates Caleb, Kevin, and ZZ for making every day fun, my roommates Allie and Jessica, and my friend Karthik for all the good times. I finally want to thank Barath for his love and support.

DEDICATION

To family and friends.

Chapter 1

INTRODUCTION

1.1 A brief history of optimization

Optimization has a long history, dating back at least several thousand years to Queen Dido in ancient Greece. While fleeing from her brother Pygmalion, she and some followers arrived at the North African coast, where she asked the local king Iarbas for a small bit of land for temporary refuge. The king granted her any slot of land that can be enclosed by an oxhide. By slicing the oxhide into thin strips and arranging the strips into a circle, she was able to cover a nearby hill. To commemerate Queen Dido, the problem of enclosing the maximum area with fixed boundary is often called the "Dido problem" in modern calculus of variations.

The brachistochrone problem is another early problem in calculus of variations. In the late 1600s, Newton was challenged to find the curve on which a frictionless bead under the influence of gravity will slide the fastest. There is a tradeoff between making the curve as short as possible (in which case, the solution is simply a straight line) and making the curve steep so the bead accelerates. One day after Newton was challenged, he showed that the ideal curve is a segment of a cycloid.

Modern optimization started around World War II, with the advent of linear programming. In 1939, the Soviet economist Leonid Kantorovich formulated a linear program to plan expenditures and reduce army costs. Less than a decade later, George Dantzig independently developed linear programming while working on planning problems for the US Air Force. Dantzig also proposed the simplex method as a way to efficiently solve linear programs, which allowed linear programs to be applied to many industries after the war.

Despite enormous progress in the 20th century, general optimization remains intractable. Perhaps the most famous example of this intractability is the traveling salesman problem. Mentioned in a salesman handbook in the early 1800s, the problem asks to find the shortest route between cities, starting and ending in the same city, so that each city is visited exactly once. The problem received much attention throughout history, and in the 1970s, Richard Karp showed that the problem is NP-complete, meaning it likely cannot be solved in polynomial time. Despite the difficulty of optimization in general, convex optimization is

computationally tractable and more popular than ever.

Boyd and Vandenberghe attribute the rise of convex optimization to two factors [19]. First, it was discovered that interior point methods, developed in the 1980s to solve linear programs, can solve certain classes of convex optimization problems almost as fast as linear programs. Indeed, Rockafellar and Wets remark that "in the study of maximization and minimization, the division between problems of convex or nonconvex type is as significant as the division in other areas of mathematics between problems of linear or nonlinear type [79, Chapter 2]." The second factor is the discovery that convex optimization problems are more common in practice that previously thought, with applications in statistics, finance, and communications emerging since the 1990s.

1.2 Outline

The introduction is organized as follows. First, convexity, convex optimization, and duality are reviewed. Several applications of convex optimization are then presented. Finally, different algorithms for approaching smooth convex optimization are discussed.

1.3 Convex optimization

1.3.1 Convexity

Convexity is most easily grasped geometrically. A subset $C \subset \mathbb{R}^n$ is convex if the line segment between any two points in C lies in C. More formally, we say that C is convex if for any two points $x, y \in C$, the points $tx + (1-t)y$ belong to C for every $t \in [0,1]$. Figure 1.1 contains several examples of convex and nonconvex sets. The core of a set C, written $\operatorname{core}(C)$, is the set of all points x in C such that for any direction d, $x + td$ lies in C for small t. We let $\operatorname{int} C$ denote the interior of C.

Let $\bar{\mathbb{R}}$ denote the extended real numbers $\mathbb{R} \cup \{\pm\infty\}$. The domain of an extended-value function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is the set

$$\operatorname{dom} f = \{x \in \mathbb{R}^n : f(x) < \infty\}.$$

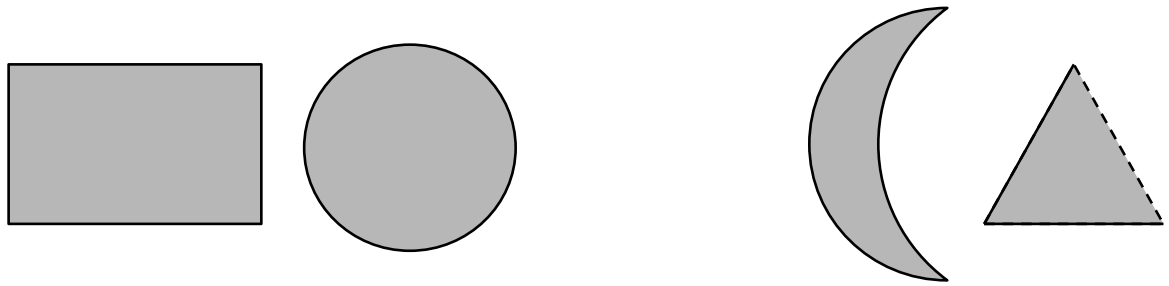


Figure 1.1: Two examples of convex sets (left) and nonconvex sets (right).

A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is convex if the inequality

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

holds for any $x, y \in \mathbb{R}^n$ and any $\theta \in [0,1]$. If strict inequality holds for distinct points $x, y \in \operatorname{dom} f$, we say f is strictly convex. Throughout we assume f is proper, meaning f is never $-\infty$ and is not always $+\infty$. Geometrically, f is convex if its epigraph (the region above its graph)

$$\operatorname{epi} f = \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : y \geq f(x)\}$$

is a convex set. In the same vein, we say that f is closed if its epigraph is a closed set. Figure 1.2 contains some examples of convex and nonconvex functions. We say f is concave if $-f$ is convex.

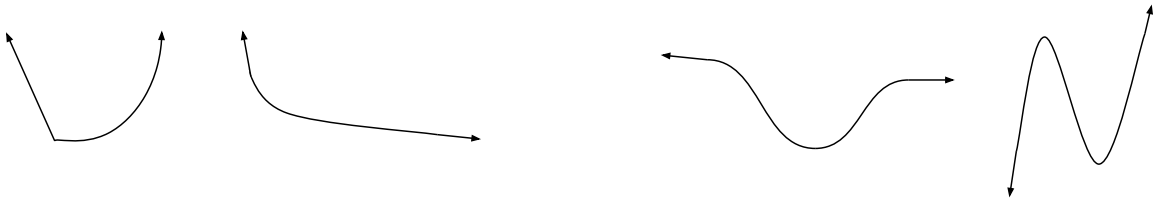


Figure 1.2: Two examples of convex functions (left) and nonconvex functions (right).

We say f is β-smooth if f is differentiable on the interior of its domain with β Lipschitz

continuous gradient. If f is convex, β-smoothness is equivalent to the inequality

$$f(y) \leq f(x) + \langle \nabla f(x),\, y - x \rangle + \frac{\beta}{2} \|y - x\|_2^2$$

holding for all $x, y \in \operatorname{dom} f$. Similarly, we say a smooth function f is α-convex if the inequality

$$f(x) + \langle \nabla f(x),\, y - x \rangle + \frac{\alpha}{2} \|y - x\|_2^2 \leq f(y)$$

holds for all $x, y \in \operatorname{dom} f$.

1.3.2 Convex optimization

The aim of convex optimization is to minimize an extended-value convex function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$. Note that we can write constrained minimization of a convex function f over a convex set C

$$\min_{x \in C} f(x)$$

as unconstrained minimization of the extended-value convex function $f(x) + \delta_C(x)$, where δ_C is the indicator function

$$\delta_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}.$$

Given a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, the convex conjugate is defined as

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \{\langle y,\, x \rangle - f(x)\}.$$

If f is closed and convex, then $(f^*)^* = f$. Convex conjugates are important in the study of Fenchel-Rockafellar duality.

Theorem 1 (Duality)**.** *[17, Theorem 3.3.5] Consider functions $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, $g : \mathbb{R}^m \to \bar{\mathbb{R}}$, and a linear map $A : \mathbb{R}^n \to \mathbb{R}^m$. Define*

$$p = \inf_x \{f(x) + g(Ax)\}$$
$$d = \sup_y \{-f^*(A^*y) - g^*(-y)\},$$

where A^* is the adjoint linear mapping. Then $p \geq d$ (weak duality) and if f and g are convex and satisfy

$$0 \in \text{int}(\text{dom}\, g - A\text{dom}\, f),$$

then $p = d$, and the supremum d is attained if finite. Roughly speaking, the condition $0 \in \text{int}(\text{dom}\, g - A\text{dom}\, f)$ means that $\text{dom}\, g$ intersects $A\text{dom}\, f$, even if one of the sets is wiggled a bit.

Before we discuss algorithms for tackling convex optimization problems, we illustrate some examples of convex optimization.

Logistic regression

Logistic regression is a powerful and popular method for predicting a binary outcome, and has been used to predict the probability that a user will click on an online ad, the probability that a voter will choose a specific candidate, and the probability that a borrower will default on a loan. In logistic regression, each observation consists of a feature vector $x \in \mathbb{R}^n$ and a binary outcome $Y \in \{0, 1\}$. In the voter example, the feature vector can encode information like the voter's age, sex, race, home city, party affiliation, income, and whether they voted in the last election. The outcome is whether the person will vote for the candidate ($Y = 1$) or not ($Y = 0$).

In logistic regression, the log-odds is modelled as a linear function of the features:

$$\log \frac{P(Y = 1|x)}{P(Y = 0|x)} = w_0 + w^T x.$$

Solving for $\log P(Y = 1|x)$ (while noting that $P(Y = 0|x) = 1 - P(Y = 1|x)$) gives

$$\log P(Y = 1|x) = -\log \left(1 + e^{-(w_0 + w^T x)}\right).$$

In a similar manner, we can compute

$$\log P(Y = 0|x) = -\log \left(1 + e^{w_0 + w^T x}\right).$$

The previous two expressions can be combined into a unified log-likelihood equation

$$\log P(Y = y|x) = -\log\left(1 + e^{(1-2y)(w_0+w^T x)}\right).$$

From the last expression, we see that once the model parameters (w_0, w) are specified, we can predict the probability of the outcome Y from the feature vector x.

Using maximum likelihood estimation to find the parameters amounts to minimizing a convex function. Suppose (x_i, y_i), for $i \in \{1, \ldots, N\}$, are independent observations. We then choose (w_0, w) to minimize the expected negative log likelihood

$$(w_0, w) \mapsto -\frac{1}{N}\sum_{i=1}^{N} \log P(Y = y_i|x_i) = \frac{1}{N}\sum_{i=1}^{N} \log\left(1 + e^{(1-2y_i)(w_0+w^T x_i)}\right).$$

Kriging

Kriging is an interpolation technique in geostatistics, where the value of a function at some point is predicted as a weighted average of its values at nearby points. For example, suppose we measure the concentration of some toxin at several locations on the surface of the lake, and we want to predict the concentration at a new location. Intuitively the concentration at any point is similar to the concentration at nearby points, so in kriging the concentration $Z(x)$ at some location $x \in \mathbb{R}^2$ on the surface is predicted as a combination of the concentrations at nearby points $\hat{Z}(x) = \sum_{i=1}^{N} \theta_i Z(x_i)$.

The toxin concentration is modelled as a Gaussian process, meaning the concentration at any location is normally distributed, and the concentrations at any finite set of locations follow a multivariate normal distribution. In so-called ordinary kriging, two assumptions are made on the underlying Gaussian process:

- The process mean μ is constant, but unknown: $\mathbf{E}\, Z(x) = \mu$ for all locations x. This assumption is reasonable if the interpolation is done on a local scale.

- The covariance between $Z(x)$ and $Z(y)$ depends only on the distance between x and y. Let $C(r)$ denote the covariance between two concentrations whose locations

are distance r apart. The function $C(r)$ is typically found by fitting a variogram $\gamma(x, y) = \frac{1}{2}\mathbf{Var}(Z(x) - Z(y))$ to the empirical data.

The estimator $\hat{Z}(x)$ is unbiased if $\sum_{i=1}^{N} \theta_i = 1$:

$$\mathbf{E}\,\hat{Z}(x) = \sum_{i=1}^{N} \theta_i \mu = \mu \sum_{i=1}^{N} \theta_i = \mathbf{E}\,Z(x).$$

Furthermore, the expected square error of the estimator is

$$\mathbf{E}\left((\hat{Z}(x) - Z(x))^2\right) = \theta^T A \theta - 2a^T\theta + \mathbf{Var}\,Z(x),$$

where

$$A_{ij} = \mathbf{Cov}(Z(x_i), Z(x_j)) = C(\left\|x_i - x_j\right\|_2) \text{ and}$$
$$a_i = \mathbf{Cov}(Z(x_i), Z(x)) = C(\|x_i - x\|_2).$$

The unbiased estimator of the concentration with smallest square error is found by solving the quadratic program

$$\begin{aligned} \min \quad & \theta^T A \theta - 2a^T\theta \\ \text{s.t.} \quad & \sum_{i=1}^{N} \theta_i = 1. \end{aligned}$$

Lasso

Lasso is a way of fitting a parsimonious linear model in the situation where there are many more measurements than observations. To motivate lasso, suppose we want to know if certain genes affect some characteristic of a person, such as blood pressure, weight, or BMI. We study N people, and for each person we take n gene measurements and measure some response such as BMI. We store the gene measurements in a data matrix $X \in \mathbb{R}^{N\times n}$ and the response measurements in a vector $y \in \mathbb{R}^N$. The simplest model to fit is a linear one:

$$y = X\beta + \epsilon,$$

where $\beta \in \mathbb{R}^n$ are model parameters, and $\epsilon \in \mathbb{R}^N$ are errors that we minimize in the fitting process. Collecting data on people is expensive, and for each person there are a lot of genes to consider ($N \ll n$), so linear regression is ill-posed. Apriori we suspect that only a few genes affect the response, so the idea behind lasso is to find a sparse parameter vector β with small error ϵ.

The objective we minimize in lasso is the convex loss function

$$\beta \mapsto \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 ,$$

where λ is a regularization parameter. Minimizing the one-norm promotes sparsity. In essence, lasso attempts to fit the data (by making $\frac{1}{2N} \|y - X\beta\|_2^2$ small) and choose a parsimonious model (by making $\|\beta\|_1$ small). The parameter λ controls the tradeoff between these aims.

Least absolute deviations

Least absolute deviations (LAD) is a robust alternative to traditional least squares regression. In least squares linear regression, we model

$$y = X\beta + \epsilon,$$

and we fit the model by minimizing the square error

$$\beta \mapsto \frac{1}{2} \|y - X\beta\|_2^2 .$$

The squaring of errors means that least squares regression is significantly influenced by outliers. There are two approaches to rectify this issue: either outliers must be identified and removed before fitting, or the fitting process must be insensitive to outliers; LAD is an example of the latter approach. In LAD, the estimate $\hat{\beta}$ is chosen to minimize the sum of absolute errors:

$$\beta \mapsto \|y - X\beta\|_1 .$$

Although there is no closed-form solution for $\hat{\beta}$, it is the solution to a convex, albeit nonsmooth, optimization problem.

1.4 Algorithms for convex optimization

Simply formulating a question as a convex program isn't useful unless the optimization problem can be efficiently solved. We will focus on iterative algorithms for minimizing smooth convex functions. An iterative method starts at an initial point x_0 and produces iterates $x_1, x_2, x_3, \ldots$ with $f(x_k) \to \min_{x \in \mathbb{R}^n} f(x)$ as $k \to \infty$. An algorithm is judged by how many iterations it takes to reach an ϵ-minimizer, i.e., a point x that satisfies $f(x) - \min_{x \in \mathbb{R}^n} f(x) \leq \epsilon$. From here on, we let x^* denote a minimizer of f, and we let f^* denote the optimal value, i.e.,

$$f^* = f(x^*) = \min_{x \in \mathbb{R}^n} f(x).$$

1.4.1 Interior point methods

Interior point methods were developed in the mid-1980s to solve linear programs, and were later applied to convex optimization in the 1990s. The simplest interior-point methods are "path following." Consider the following constrained convex program

$$\begin{aligned} \min. \quad & \langle c, x \rangle \\ \text{s.t.} \quad & x \in X, \end{aligned}$$

where X is a closed, convex domain. In interior-point methods, we equip X with a barrier F that is defined on $\operatorname{int} X$ and blows up on the boundary. In addition, the barrier is required to have a special "ν-self concordant property."

Definition 1. *A barrier F for X is ν-self-concordant if it is smooth, convex, and for all x in* $\operatorname{int} X$ *its derivatives satisfy*

$$|D^3 F(x)[h,h,h]| \leq 2 D^2 F(x)[h,h]^{3/2} \text{ for all } h \in \mathbb{R}^n, \text{ and}$$

$$\nabla F(x)^T \nabla^2 F(x)^{-1} \nabla F(x) \leq \nu.$$

The value ν is called the parameter of the barrier F.

Fix a barrier F for X, and consider the family of objective functions

$$F_t(x) = t\,\langle c,\, x\rangle + F(x),\ t \geq 0.$$

Under mild assumtions, each function F_t has a unique minimizer $x^*(t)$, and $x^*(t) \to x^*$ as $t \to \infty$. The curve $\{x^*(t) : t \geq 0\}$ is called the central path. A path following method "traces" the central path to the minimizer.

The primal path following interior point method roughly works as follows. We start at a point x_0 that is "near" the central path at t_0. During each successive iteration, we increment $t_{k+1} = \theta t_k$, with $\theta > 1$, and we set x_{k+1} by taking a single Newton step to minimize $F_{t_{k+1}}$ starting at x_k (i.e., we reach x_{k+1} by moving toward the minimizer of the second-order Taylor expansion of $F_{t_{k+1}}$ at x_k).

Provided the updates are done in a careful way, and the barrier has the special ν-self concordance property, the primal path following method finds a strictly feasible, ϵ-minimizer in at most

$$O(1)\ \sqrt{\nu}\ \log\left(\frac{\nu}{t_0\epsilon} + 2\right)$$

steps [67].

The outlined path following algorithm has the best known complexity bound for convex programming, but in practice it's better to follow the central path more aggressively using primal-dual methods. There are powerful primal-dual interior point methods for symmetric and asymmetric conic programming, which encompass linear programming, quadratic programming, semidefinite programming, second-order cone programming, and geometric programming. All interior point methods have complexity that depends on the barrier parameter ν. As such, discovering self-concordant barriers with small parameters is important.

In Chapter 2, we prove a conjectured self-concordant barrier for the generalized power cone, and describe some modeling applications of the new barrier. We briefly describe this work here.

Nesterov introduced the power cone (with parameter $\theta \in (0, 1)$)

$$\left\{(x, y, z) \in \mathbb{R}^2_+ \times \mathbb{R} : x^\theta y^{1-\theta} \geq |z|\right\}$$

to model constraints involving powers [71]. For example, the inequality $|y|^p \leq t$ (with $p > 1$) holds if and only if $(t, 1, y)$ lies in power cone with parameter $\theta = p^{-1}$. Nesterov constructed a 4-self-concordant barrier for the power cone [71], and in [30], Chares found an improved 3-self-concordant barrier for the power cone. In addition, Chares proposed the (n, m)-generalized power cone (with parameter $\alpha \in \Delta_n := \{x \in \mathbb{R}^n : x \geq 0,\ \sum_{i=1}^n x_i = 1\}$ in the simplex)

$$\mathcal{K}_\alpha^{(n,m)} = \left\{ (x, z) \in \mathbb{R}_+^n \times \mathbb{R}^m : \prod_{i=1}^n x_i^{\alpha_i} \geq \|z\| \right\}.$$

When $n = 2$ and $m = 1$, the generalized power cone reduces to the usual power cone.

Chares conjectured that

$$F(x, z) := -\log\left(\prod_{i=1}^n x_i^{2\alpha_i} - \|z\|^2 \right) - \sum_{i=1}^n (1 - \alpha_i) \log(x_i)$$

is an $(n+1)$-self-concordant barrier for $\mathcal{K}_\alpha^{(n,m)}$. Moreover, he proved that his proposed barrier is nearly optimal in that any self-concordant barrier for $\mathcal{K}^{(n,m)}$ has parameter at least n. In Chapter 2, we prove his conjecture, and in the process, give an $(n+1)$-self-concordant barrier for the high-dimensional nonnegative power cone (with parameter $\alpha \in \Delta_n$)

$$\mathcal{K}_\alpha^{+} = \left\{ (x, z) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \prod_{i=1}^n x_i^{\alpha_i} \geq z \right\}.$$

One application for the generalized power cone is to model the rotated positive power cone. Let $\alpha \in \Delta_m$ be in the simplex, and let $a_1, \ldots, a_m \in \mathbb{R}^n$ be nonnegative vectors. In [84], Nemirovski and Tuncel give a self-concordant barrier for the rotated positive power cone

$$\mathcal{C} = \left\{ (x, t) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \prod_{i=1}^m \langle a_i,\, x \rangle^{\alpha_i} \geq t \right\}$$

with parameter $\nu = 1 + \left(\frac{7}{3}\right)^2 n$. Using Chares' proposed barrier for the generalized power cone, we can construct an $(m+2)$-self-concordant barrier for $\mathcal{C}$ [30, Section 3.1.4]. Indeed, observe the inclusion $(x, t) \in \mathcal{C}$ holds if and only if the inclusions $(Ax, t) \in \mathcal{K}_\alpha^{(m,1)}$ and $t \in \mathbb{R}_+$ hold, where A is a matrix with rows given by the vectors a_i. We can therefore construct a barrier for $\mathcal{C}$ with parameter $m + 2$ ($m + 1$ for the constraint $(Ax, t) \in \mathcal{K}_\alpha^{(m,1)}$ and 1 for the

constraint $t \in \mathbb{R}_+$). In conclusion, the Chares' power cone approach is beneficial compared to Nemirovski's and Tuncel's barrier when $m \leq \left(\frac{7}{3}\right)^2 n - 1 \approx 5n$.

1.4.2 First-order methods

A first-order method is an iterative scheme where the iterates depend only on gradient information; this is in contrast to interior point methods, which utilize second-order derivative information. Given a starting point $x_0 \in \mathbb{R}^n$, most first-order methods generate iterates $x_1, x_2, \ldots$ in $\mathbb{R}^n$ satisfying

$$x_k \in x_0 + \operatorname{span}\left(\nabla f(x_0), \ldots, \nabla f(x_{k-1})\right) \text{ for } k \geq 1.$$

First-order methods were first explored in the 1980s, but fell out of fashion with the rise of extremely fast interior point methods in the 90s. Many recent problems in "big data" are too large for interior point methods to handle, and so modern interest in first-order methods has exploded.

An important development in the theory of optimization is the realization that for different problem classes, there is a limit on how fast first-order methods can be in the worst case [70, Chapter 2].

Theorem 2 (Complexity lower bounds)**.** *Given any first-order method, any starting point $x_0 \in \mathbb{R}^n$, and any iteration number $k \in \{1, \ldots, \frac{1}{2}(n-1)\}$, there is a β-smooth, convex function with*

$$f(x_k) - f^* \geq \frac{\beta \left\|x_0 - x^*\right\|_2^2}{8(k+1)^2}.$$

In other words, no iterative first-order method can find an ϵ-minimizer of a general β-smooth, convex function in fewer than $O(\sqrt{\beta/\epsilon})$ iterations. Similarly, there is a β-smooth, α-convex function with

$$f(x_k) - f^* \geq \frac{\alpha}{2}\left(\frac{\sqrt{\beta} - \sqrt{\alpha}}{\sqrt{\beta} + \sqrt{\alpha}}\right)^{2k} \left\|x_0 - x^*\right\|_2^2,$$

so no first-order method can find an ϵ-minimizer of a general β-smooth, α-convex function in fewer than $O(\sqrt{\beta/\alpha}\ \log(1/\epsilon))$ iterations.

The simplest first-order method is gradient descent. Gradient descent is motivated by the fact the negative gradient points in the direction of local steepest descent; its iterates are recursively defined by

$$x_k = x_{k-1} - t\nabla f(x_{k-1}),$$

where t is some fixed step-size. Although gradient descent is intuitive, a careful analysis of its complexity reveals that it is far from optimal.

For a general β-smooth, convex function, gradient descent (with step size $t = 1/\beta$) generates iterates that satisfy

$$f(x_k) - f^* \leq \frac{2\beta \|x_0 - x^*\|_2^2}{k+4}.$$

Comparing this rate with the lower complexity bound in Theorem 2, we see there is room for improvement. Similarly, if f is a β-smooth, α-convex function, the iterates of gradient descent (with step size $t = 2/(\alpha + \beta)$) satisfy

$$f(x_k) - f^* \leq \frac{\beta}{2}\left(1 - \frac{\alpha}{\beta}\right)^{2k} \|x_0 - x^*\|_2^2,$$

which is also suboptimal.

In the mid-1980s, Nesterov devised groundbreaking "accelerated" gradient methods that achieve the best possible rates in Theorem 2. FISTA, a slick version of Nesterov's original method due to Beck and Teboulle, is described in Algorithm 1 [9]. Unfortunately, the derivation of Nesterov's method relies on clever algebraic tricks, and the algorithm itself is unintuitive. With the recent renewed interest in first-order methods, researchers have tried to interpret Nesterov's method or develop alternative accelerated schemes that are intuitive.

In Chapter 3, we develop an intuitive, first-order method for minimizing β-smooth, α-convex functions. Moreover, our algorithm achieves the best theoretical rate, and can be further accelerated in practice by storing information in "memory." We briefly introduce and motivate the method here.

Consider a β-smooth, α-strongly convex function $f : \mathbb{R}^n \to \mathbb{R}$ that we want to minimize. Given any point $x \in \mathbb{R}^n$, let $x^+ := x - \frac{1}{\beta}\nabla f(x)$ denote a short step in the direction of the

Algorithm 1: Nesterov's accelerated method (FISTA version)

Input: Starting point $y_1 = x_0 \in \mathbb{R}^n$, $t_1 = 1$, and Lipschitz constant of gradient β.

for $k = 1, 2, \ldots$ **do**

$x_k = y_k - \frac{1}{\beta}\nabla f(y_k)$;

$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;

$y_{k+1} = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1})$

end

negative gradient, and let $x^{++} := x - \frac{1}{\alpha}\nabla f(x)$ denote a long step. To develop our new method, observe that every point $\bar{x}$ provides a quadratic under-estimator of the objective function, having a canonical form. Indeed, completing the square in the strong convexity inequality yields

$$f(x) \geq \left(f(\bar{x}) - \frac{\|\nabla f(\bar{x})\|_2^2}{2\alpha}\right) + \frac{\alpha}{2}\left\|x - \bar{x}^{++}\right\|_2^2. \tag{1.1}$$

Suppose we have now available two quadratic lower-estimators:

$$f(x) \geq Q_A(x) := v_A + \frac{\alpha}{2}\|x - x_A\|_2^2 \qquad \text{and} \qquad f(x) \geq Q_B(x) := v_B + \frac{\alpha}{2}\|x - x_B\|_2^2.$$

Clearly, the minimal values of Q_A and of Q_B lower-bound the minimal value of f. For any $\lambda \in [0, 1]$, the average $Q_\lambda := \lambda Q_A + (1 - \lambda)Q_B$ is again a quadratic lower-estimator of f. Thus we are led to the question:

What choice of λ yields the tightest lower-bound on the minimal value of f?

To answer this question, observe the equality

$$Q_\lambda(x) := \lambda Q_A(x) + (1 - \lambda)Q_B(x) = v_\lambda + \frac{\alpha}{2}\|x - c_\lambda\|_2^2,$$

where

$$c_\lambda = \lambda x_A + (1 - \lambda)x_B$$

and

$$v_\lambda = v_B + \left(v_A - v_B + \frac{\alpha}{2}\|x_A - x_B\|_2^2\right)\lambda - \left(\frac{\alpha}{2}\|x_A - x_B\|_2^2\right)\lambda^2. \tag{1.2}$$

In particular, the average Q_λ has the same canonical form as Q_A and Q_B. A quick computation now shows that v_λ (the minimum of Q_λ) is maximized by setting

$$\bar{\lambda} := \mathrm{proj}_{[0,1]}\left(\frac{1}{2} + \frac{v_A - v_B}{\alpha\|x_A - x_B\|_2^2}\right).$$

With this choice of λ, we call the quadratic function $\overline{Q} = \bar{v} + \frac{\alpha}{2}\|\cdot - \bar{c}\|^2$ the *optimal averaging* of Q_A and Q_B. See Figure 1.3 for an illustration.

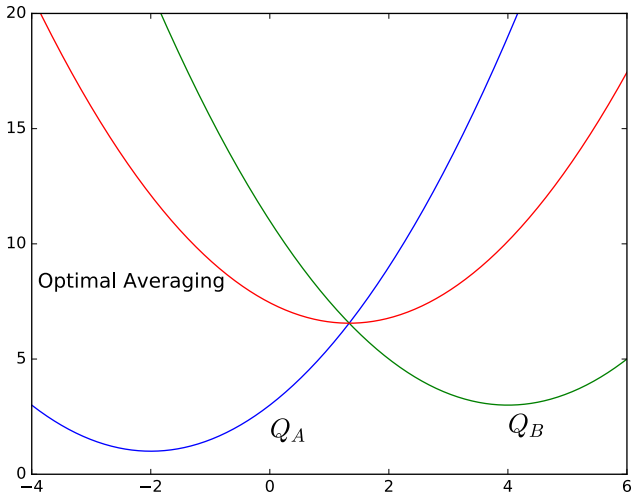


Figure 1.3: The optimal averaging of $Q_A(x) = 1 + 0.5(x+2)^2$ and $Q_B(x) = 3 + 0.5(x-4)^2$.

An algorithmic idea emerges. Given a current iterate x_k, form the quadratic lower-model $Q(\cdot)$ in (1.1) with $\bar{x} = x_k$. Then let Q_k be the optimal averaging of Q and the quadratic lower model Q_{k-1} from the previous step. Finally define x_{k+1} to be the minimizer of Q_k, and repeat. Though attractive, the scheme does not converge at an optimal rate. The main idea behind acceleration, natural in retrospect, is a separation of roles: one must maintain two sequences of points x_k and c_k. The points x_k will generate quadratic lower models as above, while c_k will be the minimizers of the quadratics. We summarize the proposed method in

Algorithm 2. Each iteration of Algorithm 2 forms an optimal average of the current lower quadratic model with the one from the previous iteration; that is, as stated the scheme has a memory size of one. We can accelerate the method further by averaging multiple quadratics in each iteration, see Chapter 3 for details.

Algorithm 2: Optimal Quadratic Averaging

Input: Starting point x_0 and strong convexity constant $\alpha > 0$.

Output: Final quadratic $Q_K(x) = v_K + \frac{\alpha}{2}\|x - c_K\|_2^2$ and x_K^+.

Set $Q_0(x) = v_0 + \frac{\alpha}{2}\|x - c_0\|_2^2$, where $v_0 = f(x_0) - \frac{\|\nabla f(x_0)\|_2^2}{2\alpha}$ and $c_0 = x_0^{++}$;

for *k = 1, . . . , K* **do**

Set x_k to be the minimizer of f over line through c_{k-1} and x_{k-1}^+ ;

Set $Q(x) = \left(f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha}\right) + \frac{\alpha}{2}\left\|x - x_k^{++}\right\|_2^2$;

Let $Q_k(x) = v_k + \frac{\alpha}{2}\|x - c_k\|_2^2$ be the optimal averaging of Q and Q_{k-1} ;

end

1.5 Reformulating convex problems

How an optimization problem is formulated determines which algorithmic tools can be used to solve it, and it isn't clear how to solve some formulations at all. In Chapter 4, we discuss a systematic method to reformulate optimization problems by interchanging the objective with one constraint function, and give many examples of where the strategy is applicable. The approach generalizes the popular SPGL1 algorithm for lasso and has important computational complexity implications.

To motivate the discussion, consider the lasso problem of recovering a sparse 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 problem formulations.

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

Computationally, BP_σ is perceived to be the most challenging of the three because of the complicated geometry of the feasible region. For example, a projected- or proximal-gradient method for LS_τ or QP_λ requires relatively little cost per iteration[1] beyond forming the product Ax or A^Ty. In contrast, a comparable first-order method for BP_σ, such as the alternating direction method of multipliers (ADMM), requires at each iteration the solution of a linear least-squares problem and maintains iterates that are both infeasible and suboptimal. Consequently, problems LS_τ and QP_λ are most often solved in practice, and most algorithm development and implementation targets these versions of the problem. Nevertheless, the formulation BP_σ is often more natural, since the parameter σ plays an entirely transparent role, signifying an acceptable tolerance on the data misfit.

In Chapter 4, we target optimization problems generalizing the formulation BP_σ. Setting the stage, consider the pair of problems

$$\min_{x\in\mathcal{X}} \quad \varphi(x) \quad \text{subject to} \quad \rho(Ax-b) \le \sigma, \tag{$\mathcal{P}_\sigma$}$$

and

$$\min_{x\in\mathcal{X}} \quad \rho(Ax-b) \quad \text{subject to} \quad \varphi(x) \le \tau, \tag{$\mathcal{Q}_\tau$}$$

where $\mathcal{X}$ is a closed convex set, φ and ρ are (possibly infinite-valued) closed convex functions, and A is a linear map. Here, $\mathcal{P}_\sigma$ and $\mathcal{Q}_\tau$ extend the problems BP_σ and LS_τ, respectively. Such formulations are ubiquitous in contemporary optimization and its applications, see Chapter 4. Our working assumption is that the level-set problem $\mathcal{Q}_\tau$ is easier to solve than $\mathcal{P}_\sigma$—perhaps because it allows for a specialized algorithm for its solution.

[1] Projection onto the ball $\{x : \|x\|_1 \le \tau\}$ requires $O(n \log n)$ operations; the proximal map for the function $\lambda\|x\|_1$ requires $O(n)$ operations.

The proposed approach, which we describe 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*:

$$\varphi(x) \leq \text{OPT} \text{ and } \rho(Ax - b) \leq \sigma + \epsilon,$$

where OPT is the optimal value of $\mathcal{P}_\sigma$. The proposed strategy is based on exchanging the roles of the objective and constraint functions in $\mathcal{P}_\sigma$, and approximately solving a sequence of level-set problems $\mathcal{Q}_\tau$ for varying parameters τ.

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

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

The univariate function v thus defined is nonincreasing and convex. Under the mild assumption that the constraint $\rho(Ax - b) \leq \sigma$ is active at any optimal solution of $\mathcal{P}_\sigma$, it is easy to see that the value $\tau_* := \text{OPT}$ satisfies the equation

$$v(\tau) = \sigma. \tag{1.4}$$

Conversely, it is immediate that for any $\tau \leq \tau_*$ satisfying $v(\tau) \leq \sigma + \epsilon$, solutions of $\mathcal{Q}_\tau$ are super-optimal and ϵ-feasible for $\mathcal{P}_\sigma$, as required. In summary, we have translated the problem $\mathcal{P}_\sigma$ to that of finding the minimal root of the nonlinear univariate equation (1.4).

Our technical assumptions on the problem $\mathcal{P}_\sigma$ are relatively few, and so in principle the approach applies to a wide class of convex optimization problems. In order to make this scheme practical, however, it is essential that approximate solutions of $\mathcal{Q}_\tau$ can be efficiently computed over a sequence of parameters τ.

Hence, efficient implementations attempt to warm start each new problem. It is thus desirable that the sequence of parameters τ_k increases monotonically, since this guarantees that the approximate solutions of $\mathcal{Q}_{\tau_k}$ are feasible for the next problem in the sequence. Bisection methods do not have this property, and we therefore propose variants of secant and Newton methods that accommodate inexact oracles for v and exhibit the desired monotonicity

property. We prove that the resulting root-finding procedures unconditionally have a global linear rate of convergence. 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).

Chapter 2

NEW BARRIERS FOR POWER CONES

Authors

Scott Roy, Lin Xiao

Abstract

Nesterov introduced the power cone, together with a 4-self-concordant barrier for it, to model constraints involving powers in convex programming. In his PhD thesis, Chares found an improved 3-self-concordant barrier for the power cone. In addition, Chares introduced the generalized power cone, and conjectured a "nearly optimal" self-concordant barrier for it. Chares numerically verified his proposed barrier was self-concordant, but was unable to prove this. In this short note, we describe some modeling applications of the generalized power cone, and prove that Chares' proposed barrier is self-concordant. As a byproduct of our analysis, we derive a self-concordant barrier for the high dimensional nonnegative power cone.

2.1 Introduction

General convex optimization can be reduced to minimization of a linear function $x \mapsto \langle c,\, x\rangle$ over a convex set X:

$$\min_{x\in X} \langle c,\, x\rangle\,.$$

In interior point methods, the constraint region X is encoded with a barrier $F : \operatorname{int} X \to \mathbb{R}$, a smooth, convex function that blows up at the boundary of X, i.e., $F(x) \to \infty$ as $x \to \partial X$. Nesterov and Nemirovski introduced *ν-self-concordant barriers* to analyze interior point methods in their ground breaking book [72].

Definition 2. *Let $X \subset \mathbb{R}^n$ be an open convex set. A function $F : X \to \mathbb{R}$ is a ν-self-concordant barrier if F blows up on the boundary of X (i.e., $F(x) \to \infty$ as $x \to \partial X$) and F is smooth, convex, and its derivatives satisfy*

$$|D^3F(x)[h,h,h]| \le 2D^2F(x)[h,h]^{3/2} \text{ for all } h \in \mathbb{R}^n, \text{ and}$$

$$\nabla F(x)^T \nabla^2 F(x)^{-1} \nabla F(x) \le \nu.$$

The value ν is called the parameter of the barrier F.

The number of iterations an interior-point method takes to solve a convex program depends on the parameter ν of the barrier used to encode the constraint region X. Researchers have thus focussed on representing constraints with "efficient" (small parameter) barriers.

Nesterov introduced the power cone (with parameter $\theta \in (0,1)$)

$$\left\{(x,y,z) \in \mathbb{R}^2_+ \times \mathbb{R} : x^\theta y^{1-\theta} \ge |z|\right\}$$

to model constraints involving powers [71]. For example, the inequality $|y|^p \le t$ (with $p > 1$) holds if and only if $(t, 1, y)$ lies in power cone with parameter $\theta = p^{-1}$. Nesterov constructed a 4-self-concordant barrier for the power cone [71], and in [30], Chares found an improved 3-self-concordant barrier for the power cone. In addition, Chares proposed the (n,m)-generalized

power cone (with parameter $\alpha \in \Delta_n := \{x \in \mathbb{R}^n : x \geq 0,\ \sum_{i=1}^n x_i = 1\}$ in the simplex)

$$\mathcal{K}_\alpha^{(n,m)} = \left\{ (x,z) \in \mathbb{R}_+^n \times \mathbb{R}^m : \prod_{i=1}^n x_i^{\alpha_i} \geq \|z\| \right\}.$$

When $n = 2$ and $m = 1$, the generalized power cone reduces to the usual power cone.

Chares conjectured that

$$F(x,z) := -\log\left(\prod_{i=1}^n x_i^{2\alpha_i} - \|z\|^2 \right) - \sum_{i=1}^n (1-\alpha_i)\log(x_i)$$

is an $(n+1)$-self-concordant barrier for $\mathcal{K}_\alpha^{(n,m)}$. Moreover, he proved that his proposed barrier is nearly optimal in that any self-concordant barrier for $\mathcal{K}^{(n,m)}$ has parameter at least n. In this short note, we prove his conjecture, and in the process, give an $(n+1)$-self-concordant barrier for the high-dimensional nonnegative power cone (with parameter $\alpha \in \Delta_n$)

$$\mathcal{K}_\alpha^{+} = \left\{ (x,z) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \prod_{i=1}^n x_i^{\alpha_i} \geq z \right\}.$$

One application for the generalized power cone is to model the rotated positive power cone. Let $\alpha \in \Delta_m$ be in the simplex, and let $a_1, \ldots, a_m \in \mathbb{R}^n$ be nonnegative vectors. In [84], Nemirovski and Tuncel give a self-concordant barrier for the rotated positive power cone

$$\mathcal{C} = \left\{ (x,t) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \prod_{i=1}^m \langle a_i,\, x\rangle^{\alpha_i} \geq t \right\}$$

with parameter $\nu = 1 + \left(\frac{7}{3}\right)^2 n$. Using Chares' proposed barrier for the generalized power cone, we can construct an $(m+2)$-self-concordant barrier for $\mathcal{C}$ [30, Section 3.1.4]. Indeed, observe the inclusion $(x,t) \in \mathcal{C}$ holds if and only if the inclusions $(Ax,t) \in \mathcal{K}_\alpha^{(m,1)}$ and $t \in \mathbb{R}_+$ hold, where A is a matrix with rows given by the vectors a_i. We can therefore construct a barrier for $\mathcal{C}$ with parameter $m+2$ ($m+1$ for the constraint $(Ax,t) \in \mathcal{K}_\alpha^{(m,1)}$ and 1 for the constraint $t \in \mathbb{R}_+$). In conclusion, the Chares' power cone approach is beneficial compared to Nemirovski's and Tuncel's barrier when $m \leq \left(\frac{7}{3}\right)^2 n - 1 \approx 5n$.

2.2 New barriers for power cones

We start by precisely stating the results we prove.

Theorem 3. *The function*

$$F(x,z) := -\log\left(\prod_{i=1}^{n} x_i^{2\alpha_i} - \|z\|^2\right) - \sum_{i=1}^{n}(1-\alpha_i)\log(x_i)$$

is an $(n+1)$-self-concordant barrier for the (n,m)-generalized power cone

$$\mathcal{K}_\alpha^{(n,m)} = \left\{(x,z) \in \mathbb{R}_+^n \times \mathbb{R}^m : \prod_{i=1}^{n} x_i^{\alpha_i} \geq \|z\|\right\}.$$

Moreover, this barrier is nearly optimal in that any self-concordant barrier for the (n,m)-generalized power cone has barrier parameter at least n.

Theorem 4. *The function*

$$F(x,z) := -\log\left(\prod_{i=1}^{n} x_i^{\alpha_i} - z\right) - \sum_{i=1}^{n}(1-\alpha_i)\log(x_i) - \log(z)$$

is an $(n+1)$-self-concordant barrier for the high-dimensional nonnegative power cone

$$\mathcal{K}_\alpha^{+} = \left\{(x,z) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \prod_{i=1}^{n} x_i^{\alpha_i} \geq z\right\}.$$

The rest of the paper is devoted to proving Theorem 3 and Theorem 4. In what follows, we let a prime denotes the total derivative at x in the direction d; so, for example, $G' = DG(x)[d]$ and $G'' = D^2G(x)[d,d]$.

Lemma 1 (Composition with logarithm)**.** *Fix a point x and direction d. Suppose that f is a positive concave function. Moreover, suppose that G is convex and satisfies $G''' \leq 2(G'')^{3/2}$. If f and G satisfy*

$$3(G'')^{1/2} f'' \leq f''' \tag{2.1}$$

then

$$F := -\log(f) + G$$

satisfies $F''' \leq 2(F'')^{3/2}$.

Proof. Let $\sigma_1 = \left(\frac{f'}{f}\right)^2$, $\sigma_2 = \frac{-f''}{f}$, and $\sigma_3 = G''$. The hypotheses imply each σ_i is nonnegative. Now simple calculations yield the following:

$$\begin{aligned}
F'' &= \sigma_1 + \sigma_2 + \sigma_3 \\
F''' &= -2\left(\frac{f'}{f}\right)^3 - 3\sigma_2\left(\frac{f'}{f}\right) - \frac{f'''}{f} + G''' \\
&\leq 2\sigma_1^{3/2} + 3\sigma_1^{1/2}\sigma_2 + 3\sigma_3^{1/2}\sigma_2 + 2\sigma_3^{3/2} \\
&= 2(\sigma_1^{1/2} + \sigma_3^{1/2})(\sigma_1 - \sigma_1^{1/2}\sigma_3^{1/2} + \sigma_3) + 3\sigma_2(\sigma_1^{1/2} + \sigma_3^{1/2}) \\
&= (\sigma_1^{1/2} + \sigma_3^{1/2})\left(3F'' - (\sigma_1^{1/2} + \sigma_2^{1/2})^2\right) \\
&\leq 2(F'')^{3/2},
\end{aligned}$$

where we used the observation that the positive maximizer of the function $t \mapsto t(3F'' - t^2)$ occurs at $t = (F'')^{1/2}$. □

Lemma 2. *Fix a dimension $n \geq 1$ and let $\Delta_n = \{w \in \mathbb{R}_+^n : \sum_{i=1}^n w_i = 1\}$ be the simplex. Suppose we have $x \in \mathbb{R}^n$ and $w \in \Delta_n$. Define the moments*

$$\begin{aligned}
s_1 &= \sum_{i \in I} w_i x_i \\
s_2 &= \sum_{i \in I} w_i x_i^2 \\
s_3 &= \sum_{i \in I} w_i x_i^3
\end{aligned}$$

and the constants

$$\begin{aligned}
e_1 &= s_1 \\
e_2 &= s_2 - s_1^2 \\
e_3 &= s_1^3 - 3s_1 s_2 + 2s_3.
\end{aligned}$$

The matrix

$$M(x, w) = \begin{bmatrix} 6e_1 + 6\left\|x\right\|_2 & -3e_2 \\ -3e_2 & e_3 + 3\left\|x\right\|_2 e_2 \end{bmatrix}$$

is positive semidefinite.

Proof. We first show that M is positive semidefinite if its determinant is nonnegative, and then establish $\det M$ is nonnegative by induction on n. To this end, suppose that we have $\det M \geq 0$. A symmetric matrix is positive semidefinite if all its principal minors are nonnegative, so we need to show the diagonal entries M_{11} and M_{22} are nonnegative. The entry M_{11} is nonnegative because we have

$$\begin{aligned} |e_1| &= |w \cdot x| \\ &\leq \|w\|_2 \|x\|_2 \\ &\leq \|w\|_1 \|x\|_2 \\ &= \|x\|_2 . \end{aligned}$$

If M_{11} is strictly positive, then

$$M_{22} = (9e_2^2 + \det M)/M_{11}$$

is also nonnegative. If M_{11} is zero, then we have $e_1 = -\|x\|_2$. This only happens if one x_i is negative, $w_i = 1$, and all other x_j are zero. In this case, $s_1 = x_i$, $s_2 = x_i^2$, and $s_3 = x_i^3$, so M_{22} is also zero.

We now show that $\det M$ is nonnegative by induction on n. Let $D(x, w)$ denote $\det M$, where we emphasize the dependence on x and w. The function $D(\cdot, w)$ is positively homogenous of degree 4; i.e., for $t \geq 0$ we have $D(tx, w) = t^4 D(x, w)$. We therefore assume that x lies on the sphere S^{n-1}.

When $n = 2$, a simple calculation shows that, in terms of the nonnegative variables $X_i = x_i + 1$, the determinant is

$$D(x, w) = 3a(X_1 - X_2)^2(bX_1^2 + cX_1X_2 + dX_2^2),$$

where

$$a = w_1 - w_1^2$$
$$b = w_1 + w_1^2$$
$$c = 4 + 2w_1 - 2w_1^2$$
$$d = 2 - 3w_1 + w_1^2.$$

For $w_1 \in [0, 1]$, the coefficients a, b, c, and d are all nonnegative, and thus so is D.

Now suppose we have $n \geq 3$. A simple calculation shows that

$$D(x, w) = -3s_1^4 - 12s_1^3 - 18s_1^2 + 12s_1 s_3 + 18s_2 - 9s_2^2 + 12s_3.$$

The function $D(\cdot, w)$ is continuous so it suffices to establish $D(\cdot, w)$ is nonnegative on the sphere S^{n-1} intersect $\{x \in \mathbb{R}^n : \text{all } x_i \text{ are distict}\}$. Fix a vector $x \in \mathbb{R}^n$ with district components and unit norm, and let w be the minimizer of $D(x, \cdot)$ over Δ_n. We show that $D(x, w)$ is nonnegative.

We claim that w does not have full support. Before we show this, let's first see how this completes the argument. Let J be the support of w. If we have $|J| < n$, then by induction we know that $M(x_J, w_J)$ is positive semidefinite, and therefore

$$M(x, w) = M(x_J, w_J) + \begin{bmatrix} 6(1 - \|x_J\|_2) & 0 \\ 0 & 3e_2(1 - \|x_J\|_2) \end{bmatrix}$$

is also positive semidefinite.

Now we show that w does not have full support. To the contrary, suppose that w does have full support, in which case the partials of $D(x, \cdot)$ at w are all equal. Thus we have

$$v = q_i := \frac{1}{6} \frac{\partial}{\partial w_i} D = x_i(ax_i^2 + bx_i + c),$$

where $a = 2(s_1 + 1)$, $b = 3(1 - s_2)$, $c = 2(s_3 - s_1^3 - 3s_1^2 - 3s_1)$, and v is some common value. We derive contradictions in two cases.

$n \geq 4$ The numbers x_1, x_2, x_3, and x_4 are distict roots of the cubic $t \mapsto at^3 + bt^2 + ct - v$, and therefore we have $a = b = c = v = 0$. Since $b = 0$, we have $s_2 = 1$; on the other hand, the assumption that x has distinct components and unit norm implies that s_2 is strictly less than 1.

$n = 3$ Since the q_i are equal and the x_i are distinct, we have

$$0 = \frac{(x_2 - x_3)(q_1 - q_3) - (x_1 - x_3)(q_2 - q_3)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} = a\Sigma + b,$$

where $\Sigma = x_1 + x_2 + x_3$. We get a contradiction by showing that $a\Sigma + b$ is strictly positive. For this, first observe the bound

$$\begin{aligned} a\Sigma + b &= 2\Sigma + 3 + \sum_{1=1}^{3} (2\Sigma x_i - 3x_i^2) w_i \\ &\geq \min_{i=1}^{3} \; 2\Sigma + 3 + 2\Sigma x_i - 3x_i^2 \\ &= \min_{i=1}^{3} \; (1 + x_i)(3 + 2x_1 + 2x_2 + 2x_3 - 3x_i). \end{aligned}$$

For any i, we claim both $1 + x_i$ and $3 + 2x_1 + 2x_2 + 2x_3 - 3x_i$ are strictly positive. For concreteness, we focus on the case where $i = 1$. For $z \in S^2$, we have $z_1 \geq -1$ with $z_1 = -1$ if and only if $z = (-1, 0, 0)$. Thus we have $1 + x_1 > 0$ since we assumed $x_2 \neq x_3$. Similarly, the affine function $z \mapsto 3 + \begin{bmatrix} -1 & 2 & 2 \end{bmatrix}^T z$ has unique minimizer over S^2 at $z = \frac{-1}{3}(-1, 2, 2)$ with minimum value 0, and so we have $3 - x_1 + 2x_2 + 2x_3 > 0$ since we assumed $x_2 \neq x_3$.

□

Proof of Theorem 3. Any self-concordant barrier for $\mathcal{K}_\alpha^{(n,m)}$ has parameter at least n by [30, Lemma 3.1.4]. The function F is $(n+1)$-logarithmically homogeneous, so the only difficulty is showing self-concordance. Define $\xi = \prod_{i=1}^n x_i^{\alpha_i}$, $f = \xi - \frac{\|z\|^2}{\xi}$, and $G = -\sum_{i=1}^n \log(x_i)$. The proposed barrier is then

$$F = -\log(f) + G$$

and we can show self-concordance by establishing Inequality 2.1 and appealing to Lemma 1. The derivatives of ξ at x in direction Δx are

$$\begin{aligned}\xi' &= e_1\xi =: s_1\xi \\ \xi'' &= -e_2\xi =: -(s_2 - s_1^2)\xi \\ \xi''' &= e_3\xi =: (s_1^3 - 3s_1s_2 + 2s_3)\xi,\end{aligned}$$

where $s_j = \sum_{i=1}^n \alpha_i \delta_i^j$ and $\delta_i = \frac{\Delta x_i}{x_i}$. The derivatives of f at (x, z) in direction $(\Delta x, \Delta z)$ are

$$\begin{aligned}f' &= \xi' + \frac{1}{\xi}\left(e_1 \|z\|^2 - 2z \cdot \Delta z\right) \\ f'' &= \xi'' - \frac{e_2}{\xi}\|z\|^2 - \frac{2}{\xi}\|e_1 z - \Delta z\|^2 \\ f''' &= \xi''' + \frac{e_3}{\xi}\|z\|^2 + \frac{6}{\xi}\left[e_1\|e_1 z - \Delta z\|^2 + e_2 z \cdot (e_1 z - \Delta z)\right].\end{aligned}$$

Let

$$g := (G'')^{1/2} = \sqrt{\sum_{i=1}^n \delta_i^2}.$$

Inequality 2.1 is equivalent to nonnegativity of

$$f''' - 3gf'' = \underbrace{\xi''' - 3g\xi''}_{A} + \frac{1}{\xi}\underbrace{\left[(6e_1 + 6g)\|e_1 z - \Delta z\|^2 + 6e_2 z \cdot (e_1 z - \Delta z) + (e_3 + 3ge_2)\|z\|^2\right]}_{B}.$$

We show that both A and B are nonnegative. We can write $A = \xi(e_3 + 3ge_2)$, and because e_2 is nonnegative, Cauchy-Schwarz yields a lower bound on B:

$$B \geq \begin{bmatrix}\|e_1 z - \Delta z\| & \|z\|\end{bmatrix}\begin{bmatrix}6e_1 + 6g & -3e_2 \\ -3e_2 & e_3 + 3ge_2\end{bmatrix}\begin{bmatrix}\|e_1 z - \Delta z\| \\ \|z\|\end{bmatrix}.$$

As ξ is positive, nonnegativity of A and B follow from Lemma 2. □

Proof of Theorem 4. The proposed barrier is

$$\begin{aligned}F(x, w) &= -\log(\xi - z) + \log(\xi) - \sum_{i=1}^n \log(x_i) - \log(z) \\ &= -\log(f) + G,\end{aligned}$$

where $f = z - \frac{z^2}{\xi}$ and $G = -\sum_{i=1}^{m} \log(x_i)$. By Lemma 1, it suffices to show

$$f''' - 3gf''$$

is nonnegative. The derivatives of ξ at x in direction Δx are

$$\begin{aligned}
\xi' &= e_1\xi =: s_1\xi \\
\xi'' &= -e_2\xi =: -(s_2 - s_1^2)\xi \\
\xi''' &= e_3\xi =: (s_1^3 - 3s_1s_2 + 2s_3)\xi,
\end{aligned}$$

where $s_j = \sum_{i=1}^{n} \alpha_i \delta_i^j$ and $\delta_i = \frac{\Delta x_i}{x_i}$. The derivatives of f at (x, z) in direction $(\Delta x, \Delta z)$ are

$$\begin{aligned}
f' &= \frac{\xi' z^2}{\xi^2} - \frac{2z\Delta z}{\xi} \\
f'' &= \frac{-1}{\xi}\left(2(e_1 z - \Delta z)^2 + e_2 z^2\right) \\
f''' &= \frac{1}{\xi}\left(e_3 z^2 + 6e_1(e_1 z - \Delta z)^2 + 6e_2 z(e_1 z - \Delta z)\right).
\end{aligned}$$

Let

$$g := (G'')^{1/2} = \sqrt{\sum_{i=1}^{n} \delta_i^2}.$$

We must show

$$f''' - 3(G'')^{1/2} f'' = \frac{1}{\xi}\left(6\left(e_1 + g\right)(e_1 z - \Delta z)^2 + 6e_2 z(e_1 z - \Delta z) + \left(e_3 + 3ge_2\right) z^2\right),$$

a quadratic form in z and $e_1 z - \Delta_z$, is nonnegative. It suffices to note the matrix

$$M := \begin{bmatrix} 6\left(e_1 + g\right) & 3e_2 \\ 3e_2 & e_3 + 3ge_2 \end{bmatrix}$$

is positive semidefinite by Lemma 2. (The off-diagonal entries of M are negated in Lemma 2, but this does not affect positive-semidefiniteness.)

□

Chapter 3

AN OPTIMAL FIRST ORDER METHOD BASED ON OPTIMAL QUADRATIC AVERAGING

Authors

Dmitriy Drusvyatskiy, Maryam Fazel, Scott Roy

Abstract

In a recent paper, Bubeck, Lee, and Singh introduced a new first-order method for minimizing smooth strongly convex functions. Their geometric descent algorithm, largely inspired by the ellipsoid method, enjoys the optimal linear rate of convergence. We show that the same iterate sequence is generated by a scheme that in each iteration computes an optimal average of quadratic lower-models of the function. Indeed, the minimum of the averaged quadratic approaches the true minimum at an optimal rate. This intuitive viewpoint reveals clear connections to the original fast-gradient methods and cutting plane ideas, and leads to limited-memory extensions with improved performance.

3.1 Introduction

Consider a function $f\colon \mathbb{R}^n \to \mathbb{R}$ that is β-smooth and α-strongly convex. Thus each point x yields a quadratic upper estimator and a quadratic lower estimator of the function. Namely, inequalities $q(y;x) \le f(y) \le Q(y;x)$ hold for all $x, y \in \mathbb{R}^n$, where we set

$$q(y;x) := f(x) + \langle \nabla f(x),\, y - x\rangle + \frac{\alpha}{2}\,\|y-x\|_2^2\,,$$
$$Q(y;x) := f(x) + \langle \nabla f(x),\, y - x\rangle + \frac{\beta}{2}\,\|y-x\|_2^2\,.$$

Classically, one step of the steepest descent algorithm decreases the squared distance of the iterate to the minimizer of f by the fraction $1 - \alpha/\beta$. This linear convergence rate is suboptimal from a computational complexity viewpoint. Optimal first-order methods, originating in Nesterov's work [69] achieve the superior (and the best possible) linear rate $1 - \sqrt{\alpha/\beta}$; see also the discussion in [70, Section 2.2]. Such accelerated schemes, on the other hand, are notoriously difficult to analyze. Numerous recent papers (e.g. [2, 5, 23, 56, 80]) have aimed to shed new light on optimal algorithms.

This manuscript is motivated by the novel geometric descent algorithm of Bubeck, Lee, and Singh [23]. Their scheme is highly geometric, sharing some aspects with the ellipsoid method, and it achieves the optimal linear rate of convergence. Moreover, the geometric descent algorithm often has much better practical performance than accelerated gradient methods; see the discussion in [23]. Motivated by their work, in this paper we propose an intuitive method that maintains a quadratic lower model of the objective function, whose minimal value converges to the true minimum at an optimal linear rate. We will show that the two methods are indeed equivalent in the sense that they produce the same iterate sequence. The quadratic averaging viewpoint, however, has important advantages. First, it immediately yields a comparison with the original accelerated gradient method [69, 70] and cutting plane techniques. Secondly, quadratic averaging motivates a simple strategy for significantly accelerating the method in practice by utilizing accumulated information – a limited memory version of the scheme.

The outline of the paper is as follows. In Section 3.2, we describe the optimal quadratic averaging framework (Algorithm 3) – the focal point of the manuscript. In Section 3.3, we propose a limited memory version of Algorithm 3, based on iteratively solving small dimensional quadratic programs. In Section 3.4, we show that our Algorithm 3 and the geometric descent method of [23] produce the same iterate sequence. Section 3.5 is devoted to numerical illustrations, in particular showing that the optimal quadratic averaging algorithm with memory can be competitive with L-BFGS. We finish the paper with Section 3.6, where we discuss the challenges that must be overcome in order to derive proximal extensions. In the final stages of revising this paper, a new manuscript [31] appeared explaining how to overcome exactly these challenges.

3.1.1 Notation

We follow the notation of [23]. Given a point $x \in \mathbb{R}^n$, we define a *short step*

$$x^+ := x - \frac{1}{\beta}\nabla f(x)$$

and a *long step*

$$x^{++} := x - \frac{1}{\alpha}\nabla f(x).$$

Setting $y = x^+$ in the quadratic bound $f(y) \leq Q(y; x)$ yields the standard inequality

$$f(x^+) + \frac{1}{2\beta}\left\|\nabla f(x)\right\|_2^2 \leq f(x). \tag{3.1}$$

We denote the unique minimizer of f by x^*, its minimal value by f^*, and its condition number by $\kappa := \beta/\alpha$. Throughout, the symbol $B(x, R^2)$ stands for the Euclidean ball of radius R around x. For any points $x, y \in \mathbb{R}^n$, we let `line_search` (x, y) be the minimizer of f on the line between x and y.

3.2 Optimal quadratic averaging

The starting point for our development is the elementary observation that every point $\bar{x}$ provides a quadratic under-estimator of the objective function, having a canonical form.

Indeed, completing the square in the strong convexity inequality $f(x) \geq q(x; \bar{x})$ yields

$$f(x) \geq \left(f(\bar{x}) - \frac{\|\nabla f(\bar{x})\|_2^2}{2\alpha} \right) + \frac{\alpha}{2} \left\| x - \bar{x}^{++} \right\|_2^2. \tag{3.2}$$

Suppose we have now available two quadratic lower-estimators:

$$f(x) \geq Q_A(x) := v_A + \frac{\alpha}{2} \|x - x_A\|_2^2 \qquad \text{and} \qquad f(x) \geq Q_B(x) := v_B + \frac{\alpha}{2} \|x - x_B\|_2^2 .$$

Clearly, the minimal values of Q_A and of Q_B lower-bound the minimal value of f. For any $\lambda \in [0, 1]$, the average $Q_\lambda := \lambda Q_A + (1 - \lambda) Q_B$ is again a quadratic lower-estimator of f. Thus we are led to the question:

What choice of λ yields the tightest lower-bound on the minimal value of f?

To answer this question, observe the equality

$$Q_\lambda(x) := \lambda Q_A(x) + (1 - \lambda) Q_B(x) = v_\lambda + \frac{\alpha}{2} \|x - c_\lambda\|_2^2 ,$$

where

$$c_\lambda = \lambda x_A + (1 - \lambda) x_B$$

and

$$v_\lambda = v_B + \left(v_A - v_B + \frac{\alpha}{2} \|x_A - x_B\|_2^2 \right) \lambda - \left(\frac{\alpha}{2} \|x_A - x_B\|_2^2 \right) \lambda^2. \tag{3.3}$$

In particular, the average Q_λ has the same canonical form as Q_A and Q_B. A quick computation now shows that v_λ (the minimum of Q_λ) is maximized by setting

$$\bar{\lambda} := \mathrm{proj}_{[0,1]} \left(\frac{1}{2} + \frac{v_A - v_B}{\alpha \|x_A - x_B\|_2^2} \right).$$

With this choice of λ, we call the quadratic function $\overline{Q} = \bar{v} + \frac{\alpha}{2} \| \cdot - \bar{c}\|^2$ the *optimal averaging* of Q_A and Q_B. See Figure 3.1 for an illustration.

An algorithmic idea emerges. Given a current iterate x_k, form the quadratic lower-model $Q(\cdot)$ in (3.2) with $\bar{x} = x_k$. Then let Q_k be the optimal averaging of Q and the quadratic lower model Q_{k-1} from the previous step. Finally define x_{k+1} to be the minimizer of Q_k,

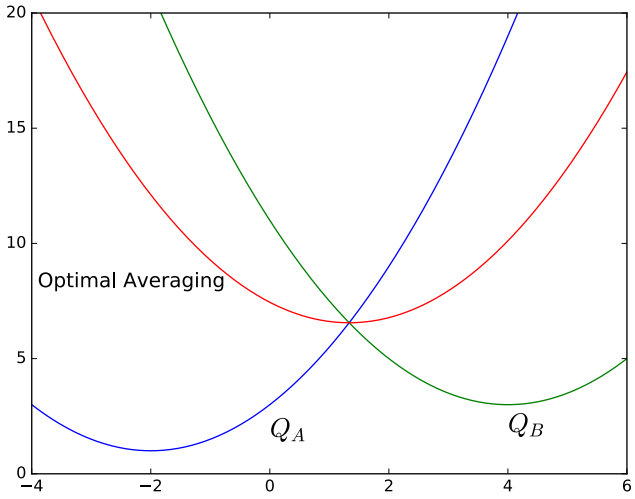


Figure 3.1: The optimal averaging of $Q_A(x) = 1 + 0.5(x+2)^2$ and $Q_B(x) = 3 + 0.5(x-4)^2$.

and repeat. Though attractive, the scheme does not converge at an optimal rate. Indeed, this algorithm is closely related to the suboptimal method in [23]; see Section 3.4.1 for a discussion. The main idea behind acceleration, natural in retrospect, is a separation of roles: one must maintain two sequences of points x_k and c_k. The points x_k will generate quadratic lower models as above, while c_k will be the minimizers of the quadratics. We summarize the proposed method in Algorithm 3. The rule for determining the iterate x_k by a line search is entirely motivated by the geometric descent method in [23].

Remark 1. When implementing Algorithm 3, we set $x_k^+ = \texttt{line\_search}\,(x_k, x_k - \nabla f(x_k))$. This does not impact the analysis as x_k^+ still satisfies the key inequality (3.1). With this modification, the algorithm does not require β as part of the input, and we have observed that the algorithm performs better numerically.

To aid in the analysis of the scheme, we record the following easy observation.

Lemma 3. *Suppose that* $\overline{Q} = \bar{v} + \frac{\alpha}{2}\|\cdot - \bar{c}\|^2$ *is the optimal averaging of the quadratics* $Q_A = v_A + \frac{\alpha}{2}\|\cdot - x_A\|^2$ *and* $Q_B = v_B + \frac{\alpha}{2}\|\cdot - x_B\|^2$. *Then the quantity* $\bar{v}$ *is nondecreasing in*

Algorithm 3: Optimal Quadratic Averaging

Input: Starting point x_0 and strong convexity constant $\alpha > 0$.

Output: Final quadratic $Q_K(x) = v_K + \frac{\alpha}{2}\|x - c_K\|_2^2$ and x_K^+.

Set $Q_0(x) = v_0 + \frac{\alpha}{2}\|x - c_0\|_2^2$, where $v_0 = f(x_0) - \frac{\|\nabla f(x_0)\|_2^2}{2\alpha}$ and $c_0 = x_0^{++}$;

for *k = 1, . . . , K* **do**

Set $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$;

Set $Q(x) = \left(f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha}\right) + \frac{\alpha}{2}\left\|x - x_k^{++}\right\|_2^2$;

Let $Q_k(x) = v_k + \frac{\alpha}{2}\|x - c_k\|^2$ be the optimal averaging of Q and Q_{k-1} ;

end

both v_A and v_B. Moreover, whenever the inequality $|v_A - v_B| \le \frac{\alpha}{2}\|x_A - x_B\|^2$ holds, we have

$$\bar{v} = \frac{\alpha}{8}\|x_A - x_B\|^2 + \frac{1}{2}(v_A + v_B) + \frac{1}{2\alpha}\left(\frac{v_A - v_B}{\|x_A - x_B\|}\right)^2.$$

Proof. Define $\hat{\lambda} := \frac{1}{2} + \frac{v_A - v_B}{\alpha\|x_A - x_B\|_2^2}$. Notice that we have

$$\hat{\lambda} \in [0,1] \quad \text{if and only if} \quad |v_A - v_B| \le \frac{\alpha}{2}\|x_A - x_B\|^2.$$

If $\hat{\lambda}$ lies in $[0,1]$, equality $\bar{\lambda} = \hat{\lambda}$ holds, and then from (3.3) we deduce

$$\bar{v} = v_{\bar{\lambda}} = \frac{\alpha}{8}\|x_A - x_B\|^2 + \frac{1}{2}(v_A + v_B) + \frac{1}{2\alpha}\left(\frac{v_A - v_B}{\|x_A - x_B\|}\right)^2.$$

If $\hat{\lambda}$ does not lie in $[0,1]$, then an easy argument shows that $\bar{v}$ is linear in v_A either with slope one or zero. If $\hat{\lambda}$ lies in $(0,1)$, then we compute

$$\frac{\partial \bar{v}}{\partial v_A} = \frac{1}{2} + \frac{1}{\alpha\|x_A - x_B\|_2^2}(v_A - v_B),$$

which is nonnegative because $\frac{|v_A - v_B|}{\alpha\|x_A - x_B\|_2^2} \le \frac{1}{2}$. Since $\bar{v}$ is clearly continuous, it follows that $\bar{v}$ is nondecreasing in v_A, and by symmetry also in v_B. □

We now show that Algorithm 3 achieves the optimal linear rate of convergence.

Theorem 5 (Convergence of optimal quadratic averaging)**.** *In Algorithm 3, for every index $k \geq 0$, the inequalities $v_k \leq f^* \leq f(x_k^+)$ hold and we have*

$$f(x_k^+) - v_k \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)^k (f(x_0^+) - v_0).$$

Proof. Since in each iteration, the algorithm only averages quadratic minorants of f, the inequalities $v_k \leq f^* \leq f(x_k^+)$ hold for every index k. Set $r_0 = \frac{2}{\alpha}(f(x_0^+) - v_0)$ and define the quantities $r_k := \left(1 - \frac{1}{\sqrt{\kappa}}\right)^k r_0$. We will show by induction that the inequality $v_k \geq f(x_k^+) - \frac{\alpha}{2} r_k$ holds for all $k \geq 0$. The base case $k = 0$ is immediate, and so assume we have

$$v_{k-1} \geq f(x_{k-1}^+) - \frac{\alpha}{2} r_{k-1}$$

for some index $k - 1$. Next set $v_A := f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha}$ and $v_B := v_{k-1}$. Then the function

$$Q_k(x) = v_k + \frac{\alpha}{2} \|x - c_k\|_2^2,$$

is the optimal averaging of $Q_A(x) = v_A + \frac{\alpha}{2} \left\|x - x_k^{++}\right\|_2^2$ and $Q_B(x) = v_B + \frac{\alpha}{2} \|x - c_{k-1}\|_2^2$. An application of (3.1) yields the lower bound $\hat{v}_A$ on v_A:

$$v_A = f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha} \geq f(x_k^+) - \frac{\alpha}{2} \frac{\|\nabla f(x_k)\|_2^2}{\alpha^2} \left(1 - \frac{1}{\kappa}\right) := \hat{v}_A.$$

The induction hypothesis and the choice of x_k yield a lower bound $\hat{v}_B$ on v_B:

$$\begin{aligned} v_B \geq f(x_{k-1}^+) - \frac{\alpha}{2} r_{k-1} &\geq f(x_k) - \frac{\alpha}{2} r_{k-1} \\ &\geq f(x_k^+) + \frac{1}{2\beta} \|\nabla f(x_k)\|_2^2 - \frac{\alpha}{2} r_{k-1} \\ &= f(x_k^+) - \frac{\alpha}{2} \left(r_{k-1} - \frac{1}{\alpha^2 \kappa} \|\nabla f(x_k)\|_2^2\right) := \hat{v}_B. \end{aligned}$$

Define the quantities $d := \left\|x_k^{++} - c_{k-1}\right\|_2$ and $h := \frac{\|\nabla f(x_k)\|_2}{\alpha}$. We now split the proof into

two cases. First assume $h^2 \leq \frac{r_{k-1}}{2}$. Then we deduce

$$\begin{aligned} v_k \geq v_A \geq \hat{v}_A &= f(x_k^+) - \frac{\alpha}{2}h^2\left(1 - \frac{1}{\kappa}\right) \\ &\geq f(x_k^+) - \frac{\alpha}{2}r_{k-1}\left(\frac{1-\frac{1}{\kappa}}{2}\right) \\ &\geq f(x_k^+) - \frac{\alpha}{2}r_{k-1}\left(1 - \frac{1}{\sqrt{\kappa}}\right) \\ &= f(x_k^+) - \frac{\alpha}{2}r_k, \end{aligned}$$

where the third line follows since $2/\sqrt{\kappa} \leq 1 + 1/\kappa$ holds. Hence in this case, the proof is complete.

Next suppose $h^2 > \frac{r_{k-1}}{2}$ and let $v + \frac{\alpha}{2}\|\cdot - c\|^2$ be the optimal average of the two quadratics $\hat{v}_A + \frac{\alpha}{2}\|\cdot - x_k^{++}\|^2$ and $\hat{v}_B + \frac{\alpha}{2}\|\cdot - c_{k-1}\|^2$. By Lemma 3, the inequality $v_k \geq v$ holds. We claim that equality

$$v = \hat{v}_B + \frac{\alpha}{8}\,\frac{\left(d^2 + \frac{2}{\alpha}(\hat{v}_A - \hat{v}_B)\right)^2}{d^2} \qquad \text{holds.} \tag{3.4}$$

This follows immediately from Lemma 3, once we show $\frac{1}{2} \geq \frac{|\hat{v}_A - \hat{v}_B|}{\alpha d^2}$. To this end, note first the equality $\frac{|\hat{v}_A - \hat{v}_B|}{\alpha d^2} = \frac{\left|r_{k-1} - h^2\right|}{2d^2}$. The choice $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$ ensures:

$$d^2 - h^2 = \|x_k - c_{k-1}\|^2 - \frac{2}{\alpha}\langle \nabla f(x_k), x_k - c_{k-1}\rangle = \|x_k - c_{k-1}\|^2 \geq 0.$$

Thus we have $h^2 - r_{k-1} < h^2 \leq d^2$. Finally, the assumption $h^2 > \frac{r_{k-1}}{2}$ implies

$$r_{k-1} - h^2 < \frac{r_{k-1}}{2} < h^2 \leq d^2. \tag{3.5}$$

Hence we can be sure that (3.4) holds. Plugging in $\hat{v}_A$ and $\hat{v}_B$ yields

$$v = f(x_k^+) - \frac{\alpha}{2}\left(r_{k-1} - \frac{1}{\kappa}h^2 - \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2}\right).$$

Hence the proof is complete once we show the inequality

$$r_{k-1} - \frac{1}{\kappa}h^2 - \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2} \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right) r_{k-1}.$$

After rearranging, our task simplifies to showing the inequality

$$\frac{r_{k-1}}{\sqrt{\kappa}} \leq \frac{h^2}{\kappa} + \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2}.$$

Taking derivatives and using inequality (3.5), one can readily verify that the right-hand-side is nondecreasing in d^2 on the interval $d^2 \in [h^2, +\infty)$. Thus plugging in the endpoint $d^2 = h^2$ we deduce

$$\frac{h^2}{\kappa} + \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2} \geq \frac{h^2}{\kappa} + \frac{r_{k-1}^2}{4h^2}.$$

Minimizing the right-hand-side over all h satisfying $h^2 \geq \frac{r_{k-1}}{2}$ yields the inequality

$$\frac{h^2}{\kappa} + \frac{r_{k-1}^2}{4h^2} \geq \frac{r_{k-1}}{\sqrt{\kappa}}.$$

The proof is complete. □

It is instructive to compare optimal averaging (Algorithm 3) with Nesterov's optimal methods in [69, 70]. For convenience, we record the optimal gradient method following [70], in Algorithm 4.

Algorithm 4: General scheme of an optimal method [Nesterov]

Input: Starting points x_0 and c_0, strong convexity constant $\alpha > 0$, smoothness parameter $\beta > 0$, and initial quadratic curvature $\gamma_0 \geq \alpha$.

Output: Final quadratic $Q_K(x) = v_K + \frac{\gamma_K}{2} \|x - c_K\|_2^2$.

Set $Q_0(x) = v_0 + \frac{\gamma_0}{2} \|x - c_0\|_2^2$, where $v_0 = f(x_0) - \frac{1}{2\beta} \|\nabla f(x_0)\|_2^2$;

for *k = 1, . . . , K* **do**

- Compute averaging parameter $\lambda_k \in (0, 1)$ from $\beta\lambda_k^2 = (1 - \lambda_k)\gamma_{k-1} + \lambda_k\alpha$;
- Set $\gamma_k = (1 - \lambda_k)\gamma_{k-1} + \lambda_k\alpha$. ;
- Set $x_k = (1 - \theta_k)c_{k-1} + \theta_k x_{k-1}^+$ where $\theta_k = \frac{\gamma_k}{\gamma_{k-1} + \lambda_k\alpha}$;
- Set $Q(x) = \left(f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha}\right) + \frac{\alpha}{2} \left\|x - x_k^{++}\right\|_2^2$;
- Let c_k be the minimizer of the quadratic $Q_k(x) = (1 - \lambda_k)Q_{k-1}(x) + \lambda_k Q(x)$;

end

/* If we set $\gamma_0 = \alpha$, then we have $\gamma_k = \alpha$, $\lambda_k = \frac{1}{\sqrt{\kappa}}$, and $\theta_k = \frac{\sqrt{\kappa}}{1+\sqrt{\kappa}}$. */

Comparing Algorithms 3 and 4, we see that

- x_k is some point on the line between c_{k-1} and x_{k-1}^+, and

- Q_k is an average of the previous quadratic Q_{k-1} and the strong convexity quadratic lower bound Q based at x_k.

As we discuss in Section 3.7, we can modify Nesterov's method so that like in optimal quadratic averaging, we set $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$ in each iteration. After this change, only two differences remain between the schemes:

- the initial quadratic Q_0 is different, and

- the averaging parameter is computed differently.

These differences, however, are fundamental. In Algorithm 3, the quadratic Q_0 lower bounds f and therefore optimal averaging makes sense; in the accelerated gradient method, Q_0 does not lower bound f, and the idea of optimal averaging does not apply.

3.3 Optimal quadratic averaging with memory

Each iteration of Algorithm 3 forms an optimal average of the current lower quadratic model with the one from the previous iteration; that is, as stated the scheme has a memory size of one. We next show how the scheme easily adapts to maintaining limited memory, i.e. by averaging multiple quadratics in each iteration. We mention in passing that the authors of [23] left open the question of efficiently speeding up their geometric descent algorithm in practice. One approach of this flavor has recently appeared in [22, Section 4]. The optimal averaging viewpoint, developed here, provides a direct and satisfying alternative. Indeed, computing the optimal average of several quadratics is easy, and amounts to solving a small dimensional quadratic optimization problem.

To see this, fix t quadratics $Q_i(x) := v_i + \frac{\alpha}{2}\|x - c_i\|_2^2$, with $i \in \{1, \ldots, t\}$, and a weight vector λ in the t-dimensional simplex $\Delta_t := \left\{x \in \mathbb{R}^t : \sum_{i=1}^t x_i = 1,\ x \geq 0\right\}$. The average

quadratic

$$Q_\lambda(x) := \sum_{i=1}^{t} \lambda_i Q_i(x)$$

maintains the same canonical form as each Q_i.

Proposition 1. *Define the matrix* $C = \begin{bmatrix} c_1 & c_2 & \dots & c_t \end{bmatrix}$ *and vector* $v = \begin{bmatrix} v_1 & v_2 & \dots & v_t \end{bmatrix}^T$. *Then we have*

$$Q_\lambda(x) = v_\lambda + \frac{\alpha}{2} \|x - c_\lambda\|_2^2 ,$$

where

$$c_\lambda = C\lambda \quad \textit{and} \quad v_\lambda = \left\langle \frac{\alpha}{2} \mathrm{diag}\,(C^T C) + v,\, \lambda \right\rangle - \frac{\alpha}{2} \|C\lambda\|_2^2 .$$

Proof. The Hessian of Q_λ is simply $\frac{\alpha}{2} I$, and therefore the quadratic $Q_\lambda(x)$ has the form

$$v_\lambda + \frac{\alpha}{2} \|x - c_\lambda\|_2^2$$

for some v_λ and c_λ. Notice that c_λ is the minimizer of Q_λ, and by differentiating, we determine that $c_\lambda = \sum_{i=1}^t \lambda_i c_i = C\lambda$. We then compute

$$\begin{aligned} v_\lambda = Q_\lambda(c_\lambda) &= \sum_{i=1}^{t} \left(\lambda_i v_i + \frac{\lambda_i \alpha}{2} \|C\lambda - c_i\|_2^2 \right) \\ &= \langle v,\, \lambda \rangle + \frac{\alpha}{2} \sum_{i=1}^{t} \lambda_i \left(\|C\lambda\|_2^2 - 2 \langle C\lambda,\, c_i \rangle + \|c_i\|_2^2 \right) \\ &= \langle v,\, \lambda \rangle + \frac{\alpha}{2} \|C\lambda\|_2^2 - \alpha \left\langle C\lambda,\, \sum_{i=1}^{t} \lambda_i c_i \right\rangle + \frac{\alpha}{2} \sum_{i=1}^{t} \lambda_i \|c_i\|_2^2 \\ &= \left\langle \frac{\alpha}{2} \mathrm{diag}\left(C^T C \right) + v,\, \lambda \right\rangle - \frac{\alpha}{2} \|C\lambda\|_2^2 . \end{aligned}$$

The proof is complete. □

Naturally, we define the *optimal averaging* of the quadratics Q_i, with $i \in \{1, 2, \dots, t\}$, to be $Q_{\bar\lambda}$, where $\bar\lambda$ is the maximizer of the concave quadratic over the simplex:

$$\min_{\lambda \in \Delta_t} \; v_\lambda = \left\langle \frac{\alpha}{2} \mathrm{diag}\left(C^T C \right) + v,\, \lambda \right\rangle - \frac{\alpha}{2} \|C\lambda\|_2^2 .$$

There is no closed form expression for $\bar{\lambda}$, but one can quickly find it by solving a quadratic program in t variables, for example by an active set method. Moreover, some thought shows that the matrix $C^T C$ can be efficiently updated if one of the centers changes; we omit the details.

We propose an optimal averaging scheme with memory in Algorithm 5. As we see in Section 3.5, the method performs well numerically. Moreover, the scheme enjoys the same convergence guarantees as Algorithm 3; that is, Theorem 5 applies to Algorithm 5, with nearly the same proof (which we omit).

Algorithm 5: Optimal Quadratic Averaging with Memory

Input: Starting point x_0, strong convexity constant $\alpha > 0$, and memory size $t \geq 1$.

Output: Final quadratic $Q_K(x) = v_K + \frac{\alpha}{2}\|x - c_K\|_2^2$ and x_K^+.

Set $Q_0(x) = v_0 + \frac{\alpha}{2}\|x - c_0\|_2^2$, where $v_0 = f(x_0) - \frac{\|\nabla f(x_0)\|_2^2}{2\alpha}$ and $c_0 = x_0^{++}$;

for *k = 1, . . . , K* **do**

Set $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$;

Set $M_k(x) = f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha} + \frac{\alpha}{2}\left\|x - x_k^{++}\right\|_2^2$;

Let $Q_k(x) := v_k + \frac{\alpha}{2}\|x - c_k\|_2^2$ be the optimal averaging of the

$k+1$ quadratics $Q_{k-1},\ M_k,\ M_{k-1},\ \ldots,\ M_1$ if $k \leq t$, or of the

$t+1$ quadratics $Q_{k-1},\ M_k,\ M_{k-1},\ \ldots,\ M_{k-t+1}$ if $k \geq t+1$;

end

The reader may notice that Algorithm 5 shows some similarity to the classical Kelley's method for minimizing nonsmooth convex functions [47]. In the simplest case of minimizing a smooth convex function f on $\mathbb{R}^n$, Kelley's method iterates the following steps

$$x_{k+1} = \operatorname*{argmin}_x \ f_k(x)$$

for the functions

$$f_k(x) := \max_{i=1,\dots,k} \{f(x_i) + \langle \nabla f(x_i), x - x_i \rangle\}.$$

In other words, the scheme iteratively minimizes the (piecewise linear) lower-models f_k of f. Coming back to the optimal averaging viewpoint, suppose that $Q_{\bar{\lambda}}$ is an optimal average of the lower-bounding quadratics Q_i, for $i = 1, \dots, k$. Then we may write

$$v_{\bar{\lambda}} = \max_{\lambda \in \Delta_k} \min_x \sum_i \lambda_i Q_i(x) = \min_x \max_{\lambda \in \Delta_k} \sum_i \lambda_i Q_i(x) = \min_x \left(\max_{i=1,\dots,k} Q_i(x) \right)$$

Thus $v_{\bar{\lambda}}$ is the minimal value of the now different lower-model, $\max_{i=1,\dots,k} Q_i$, of f. Kelley's method is known to have poor numerical performance and convergence guarantees (e.g. [70, Section 3.3.2]), while Algorithm 5 achieves the optimal linear convergence rate. This disparity is of course based on the two key distinctions: (1) using quadratic lower-models coming from strong convexity instead of linear functions, and (2) maintaining two separate sequences c_k (centers) and x_k (sources of lower model updates).

3.4 Equivalence to geometric descent

Algorithm 3 is largely motivated by the geometric descent method introduced by Bubeck, Lee, and Singh [23]. In this section, we show the two methods (Algorithm 1 and Algorithm 4) indeed generate an identical iterate sequence.

3.4.1 Suboptimal geometric descent method

The basic idea of geometric descent [23] is that for each point $x \in \mathbb{R}^n$, the strong convexity lower bound $f^* \geq q(x^*; x)$ defines a ball containing x^*:

$$x^* \in B\left(x^{++}, \frac{\|\nabla f(x)\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x) - f^*\right) \right).$$

In turn, taking into account (3.1) yields the guarantee

$$x^* \in B\left(x^{++}, \left(1 - \frac{1}{\kappa}\right) \frac{\|\nabla f(x)\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x^+) - f^*\right) \right). \tag{3.6}$$

A crude upper estimate of the radius above is obtained simply by ignoring the nonnegative term $\frac{2}{\alpha}\left(f(x^+) - f^*\right)$. The suboptimal geometric descent method proceeds as follows. Suppose we have available some ball $B\left(c_0, R_0^2\right)$ containing x^*. As discussed, the quadratic lower bound at the center c_0, namely $f^* \geq q(x^*, c_0)$, yields another ball $B\left(c_0^{++}, \left(1 - \frac{1}{\kappa}\right)\frac{\|\nabla f(c_0)\|_2^2}{\alpha^2}\right)$ containing x^*. Geometrically it is clear that the intersection of these two balls must be significantly smaller than either of the individual balls. The following lemma from [23] makes this observation precise; see Figure 3.2 for an illustration.

Lemma 4 (Minimal enclosing ball of the intersection)**.** *Fix a center $x \in \mathbb{R}^n$, square radius $R^2 > 0$, step $h \in \mathbb{R}^n$, and $\epsilon \in (0, 1)$. Then there exists a new center $c \in \mathbb{R}^n$ with*

$$B\left(x, R^2\right) \cap B\left(x + h, (1 - \epsilon)\|h\|_2^2\right) \subset B\left(c, (1 - \epsilon)R^2\right).$$

An application of Lemma 4 yields a new center c_1 with

$$B\left(c_0, R_0^2\right) \cap B\left(c_0^{++}, \left(1 - \frac{1}{\kappa}\right)\frac{\|\nabla f(c_0)\|_2^2}{\alpha^2}\right) \subset B\left(c_1, \left(1 - \frac{1}{\kappa}\right)R_0^2\right).$$

Repeating the procedure with the new ball $B\left(c_1, \left(1 - \frac{1}{\kappa}\right)R_0^2\right)$ yields a sequence of centers c_k satisfying

$$\|c_k - x^*\|_2^2 \leq \left(1 - \frac{1}{\kappa}\right)^k R_0^2.$$

We note that the centers c_k and R_0^2 of the minimal enclosing balls in Lemma 4 are easy to compute; see Algorithm 1 in [23].

There is a very close connection between finding the minimal enclosing ball of the intersection of two balls and of optimally averaging quadratics. To see this, consider again two quadratics

$$f(x) \geq Q_A(x) := v_A + \frac{\alpha}{2}\|x - x_A\|_2^2 \qquad \text{and} \qquad f(x) \geq Q_B(x) := v_B + \frac{\alpha}{2}\|x - x_B\|_2^2.$$

Let $\overline{Q}$ be the optimal average of Q_A and Q_B. Notice that since Q_A, Q_B, and $\overline{Q}$ lower bound

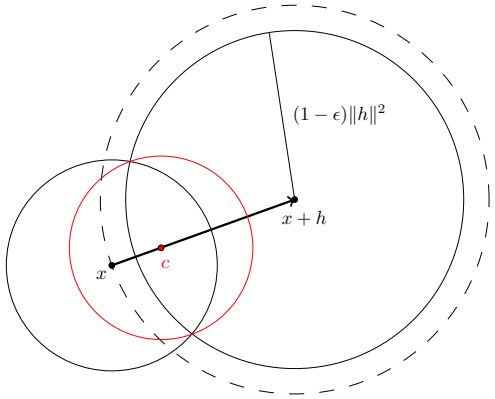


Figure 3.2: Minimal enclosing ball of the intersection.

f, the minimizer x^* of f is guaranteed to lie in the three balls:

$$\begin{aligned} B\left(x_A, R_A^2\right) \quad \text{where} \quad R_A^2 &= \frac{2}{\alpha}\left(\hat{f} - v_A\right), \\ B\left(x_B, R_B^2\right) \quad \text{where} \quad R_B^2 &= \frac{2}{\alpha}\left(\hat{f} - v_B\right), \\ B\left(\bar{c}, R^2\right) \quad \text{where} \quad R^2 &= \frac{2}{\alpha}\left(\hat{f} - \bar{v}\right), \end{aligned}$$

where $\hat{f}$ is any upper bound on f^*. We observe the following elementary fact.

Proposition 2 (Minimal enclosing ball and optimal averaging)**.** *The ball $B\left(\bar{c}, R^2\right)$ is precisely the minimal enclosing ball of the intersection $B\left(x_A, R_A^2\right) \cap B\left(x_B, R_B^2\right)$.*

Proof. Define the quantity $\hat{\lambda} = \frac{1}{2} + \frac{v_A - v_B}{\alpha \|x_A - x_B\|_2^2}$. If $\hat{\lambda}$ lies in the unit interval $[0, 1]$, then a quick computation using Lemma 3 shows the expressions

$$R^2 = R_B^2 - \frac{\left(\|x_A - x_B\|_2^2 + R_B^2 - R_A^2\right)^2}{4\|x_A - x_B\|_2^2}$$

and

$$\bar{c} = \bar{\lambda} x_A + (1 - \bar{\lambda}) x_B = \frac{1}{2}\left(x_A + x_B\right) - \frac{R_A^2 - R_B^2}{2\|x_A - x_B\|_2^2}\left(x_A - x_B\right).$$

Now observe

$$\begin{aligned} \hat{\lambda} < 0 \quad &\text{if and only if} \quad \|x_A - x_B\|_2^2 < R_A^2 - R_B^2 \\ \hat{\lambda} \in [0,1] \quad &\text{if and only if} \quad \|x_A - x_B\|_2^2 \geq \left|R_A^2 - R_B^2\right|, \text{ and} \\ \hat{\lambda} > 1 \quad &\text{if and only if} \quad \|x_A - x_B\|_2^2 < R_B^2 - R_A^2. \end{aligned}$$

Comparing with the recipe [23, Algorithm 1] for computing the minimal enclosing ball, we see that $B\left(\bar{c}, R^2\right)$ is the minimal enclosing ball of the intersection $B\left(x_A, R_A^2\right) \cap B\left(x_B, R_B^2\right)$. $\square$

3.4.2 Optimal geometric descent method

To obtain an optimal method, the authors of [23] observe that the term $\frac{2}{\alpha}\left(f(x^+) - f^*\right)$ in the inclusion (3.6) cannot be ignored. Exploiting this term will require maintaining two sequences c_k (the centers of the balls) and x_k (points for generating new balls). Suppose in iteration k, we know that x^* lies in the ball

$$B\left(c_k, R_k^2 - \frac{2}{\alpha}\left(f(x_k^+) - f^*\right)\right).$$

Consider now an arbitrary point, denoted suggestively by x_{k+1}. Then (3.6) implies the inclusion

$$x^* \in B\left(x_{k+1}^{++}, \left(1 - \frac{1}{\kappa}\right)\frac{\|\nabla f(x_{k+1})\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right). \tag{3.7}$$

If we choose x_{k+1} to satisfy $f(x_{k+1}) \leq f(x_k^+)$ and apply inequality (3.1) with $x = x_{k+1}$, we can get a new upper estimate of the initial ball,

$$x^* \in B\left(c_k, R_k^2 - \frac{1}{\kappa}\,\frac{\|\nabla f(x_{k+1})\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right). \tag{3.8}$$

It seems clear that if the centers c_k and x_{k+1}^{++} of the two balls in (3.7) and (3.8) are "sufficiently far apart", then their intersection is contained in an even smaller ball. This is the content of following lemma from [23].

Lemma 5 (Two balls shrinking)**.** *Fix centers $x_A, x_B \in \mathbb{R}^n$ and square radii $r_A^2, r_B^2 > 0$. Also fix $\epsilon \in (0,1)$ and suppose $\|x_A - x_B\|_2^2 \geq r_B^2$. Then there exists a new center $c \in \mathbb{R}^n$ such that for any $\delta > 0$, we have*

$$B\left(x_A, r_A^2 - \epsilon r_B^2 - \delta\right) \cap B\left(x_B, (1-\epsilon) r_B^2 - \delta\right) \subset B\left(c, (1-\sqrt{\epsilon}) r_A^2 - \delta\right).$$

A quick application of this result shows that provided

$$\left\|x_{k+1}^{++} - c_k\right\|_2^2 \geq \frac{\|\nabla f(x_{k+1})\|_2^2}{\alpha^2} \tag{3.9}$$

holds, there exists a new center c_{k+1} with

$$x^* \in B\left(c_{k+1}, \left(1 - \frac{1}{\sqrt{\kappa}}\right) R_k^2 - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right).$$

One way to ensure that x_{k+1} satisfies the two key conditions, $f(x_{k+1}) \leq f(x_k^+)$ and inequality (3.9), is to simply let x_{k+1} be the minimizer of f along the line between c_k and x_k^+. Trivially this guarantees the inequality $f(x_{k+1}) \leq f(x_k^+)$, while the univariate optimality condition $\nabla f(x_{k+1}) \perp (c_k - x_{k+1})$ means the triangle with vertices x_{k+1}, x_{k+1}^{++}, and c_k is a right triangle and inequality (3.9) becomes "the hypotenuse is longer than a leg." This is exactly the motivation for the line-search procedure in Algorithm 3. Repeating the process yields iterates c_k that satisfy the optimal linear rate of convergence

$$\|c_k - x^*\|_2^2 \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)^k R_0^2.$$

The precise method is described in Algorithm 6.

Remark 2. When applying an iterative method to compute $x_{k+1} = \texttt{line\_search}\left(c_k, x_k^+\right)$, one can use the following termination criterion. Check if c_k satisfies $f(c_k) \leq f(x_k^+)$, then stop and set $x_{k+1} := c_k$. Notice (3.9) holds trivially with this choice of x_{k+1}. Else stop with a trial point z on the line joining c_k and x_k^+ satisfying $f(z) \leq f(x_k^+)$ and

$$\left\|z^{++} - c_k\right\|_2^2 \geq \frac{\|\nabla f(z)\|_2^2}{\alpha^2}.$$

We claim that the line search will terminate in finite time, unless $\texttt{line\_search}\left(c_k, x_k^+\right)$ is the true minimizer of f. Indeed, since $c_k \neq \texttt{line\_search}\left(c_k, x_k^+\right)$ (otherwise we would have terminated in the if clause), one can easily check that $z = \texttt{line\_search}\left(c_k, x_k^+\right)$ satisfies the above inequality strictly.

Algorithm 6: Geometric Descent Method [Bubeck, Lee, Singh]

Input: Starting point x_0, strong convexity constant $\alpha > 0$.

Output: x_K^+

Set $c_0 = x_0^{++}$ and $R_0^2 = \frac{\|\nabla f(x_0)\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_0) - f(x_0^+)\right)$;

for *k = 1, . . . , K* **do**

Set $x_k = \texttt{line\_search}\left(x_{k-1}^+, c_{k-1}\right)$;

Set $x_A = x_k - \alpha^{-1}\nabla f(x_k)$ and $R_A^2 = \frac{\|\nabla f(x_k)\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_k) - f(x_k^+)\right)$;

Set $x_B = c_{k-1}$ and $R_B^2 = R_{k-1}^2 - \frac{2}{\alpha}\left(f(x_{k-1}^+) - f(x_k^+)\right)$;

Let $B\left(c_k, R_k^2\right)$ be the smallest enclosing ball of $B\left(x_A, R_A^2\right) \cap B\left(x_B, R_B^2\right)$;

end

The following theorem shows that Algorithm 3 and Algorithm 6 indeed produce the same iterate sequence.

Theorem 6. *Given the same initial point x_0, Algorithm 3 and Algorithm 6 produce the same iterates x_k and c_k. Moreover, we have $v_k = f(x_k^+) - \frac{\alpha}{2}R_k^2$, where v_k is the minimum value of the quadratic Q_k in Algorithm 3 and R_k is the radius of the ball in Algorithm 6.*

Proof. Let x_k and c_k denote the iterates in Algorithm 3, and let $\hat{x}_k$ and $\hat{c}_k$ be the iterates in Algorithm 6. We proceed by induction on k. It follows immediately from the definition of the algorithms that $x_0 = \hat{x}_0$, $c_0 = \hat{c}_0$, and $v_0 = f(x_0^+) - \frac{\alpha}{2}R_0^2$. Now suppose, as an inductive assumption, $x_{k-1} = \hat{x}_{k-1}$, $c_{k-1} = \hat{c}_{k-1}$, and $v_{k-1} = f(x_{k-1}^+) - \frac{\alpha}{2}R_{k-1}^2$. To see the equality $x_k = \hat{x}_k$, observe

$$x_k = \texttt{line\_search}\left(x_{k-1}^+, c_{k-1}\right) = \texttt{line\_search}\left(\hat{x}_{k-1}^+, \hat{c}_{k-1}\right) = \hat{x}_k.$$

Let $x_A = x_k^{++}$, $x_B = c_{k-1}$, $d = \|x_A - x_B\|$, and define the quantities

$$v_A = f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha}, \qquad R_A^2 = \frac{\|\nabla f(x_k)\|^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_k) - f(x_k^+)\right),$$
$$v_B = v_{k-1}, \qquad R_B^2 = R_{k-1}^2 - \frac{2}{\alpha}\left(f(x_{k-1}^+) - f(x_k^+)\right).$$

Notice that $Q_k(x) = v_k + \frac{\alpha}{2}\|x - c_k\|^2$ is the optimal averaging of $Q_A(x) := v_A + \frac{\alpha}{2}\|x - x_A\|^2$ and $Q_B(x) := v_B + \frac{\alpha}{2}\|x - x_B\|^2$, and that $B(\hat{c}_k, R_k^2)$ is the minimum enclosing ball of the intersection of $B(x_A, R_A^2)$ and $B(x_B, R_B^2)$. Simple algebra shows the relation

$$R_A^2 = \frac{2}{\alpha}\left(f(x_k^+) - v_A\right),$$

and from the inductive assumption $v_{k-1} = f(x_{k-1}^+) - \frac{\alpha}{2}R_{k-1}^2$, we also have

$$R_B^2 = \frac{2}{\alpha}\left(f(x_k^+) - v_B\right).$$

Thus, by Proposition 2 and the discussion preceding it, we have $c_k = \hat{c}_k$ and $v_k = f(x_k^+) - \frac{\alpha}{2}R_k^2$. This completes the induction. □

As we saw in Section 3.3, computing the optimal averaging of several quadratic functions is simple. On the other hand, it is far from clear how to find the minimum radius ball that encloses the intersection of more than two balls. Indeed, instead the authors of Algorithm 6 in the follow-up work [22] considered a "relaxation" that involves minimizing a self-concordant barrier for the intersection. While revising the current manuscript, we became aware that Beck in [7, Theorem 3.2] proved that the minimum enclosing ball of the intersection of finitely many balls can be computed by solving a convex quadratic program (QP). Namely, Beck showed that the squared radius of the minimal ball enclosing the intersection $\bigcap_{i=1}^t B(c_i, r_i^2)$ is exactly equal to

$$\min_{\lambda \in \Delta_t} \left\| \sum_{i=1}^t \lambda_i c_i \right\|^2 - \sum_{i=1}^t \lambda_i (\|a_i\|^2 - r_i^2),$$

provided $t \leq n-1$ and the intersection of the balls has nonempty interior. This QP is exactly the one we derived in Section 3.3 for the optimal quadratic averaging method with memory. Note that our derivation of the QP in Section 3.3 was completely elementary; the proof of [7,

Theorem 3.2], on the other hand, is much more sophisticated relying on an S-lemma-type result.

Proposition 3 (Optimal quadratic averaging & minimal enclosing ball)**.**
Let $Q(x) = v + \frac{\alpha}{2}\|x - c\|^2$ be the optimal averaging of quadratics $Q_i(x) = v_i + \frac{\alpha}{2}\|x - c_i\|^2$ for $i = 1, \ldots, t$ with $t < n$. Fix a real number $s \geq v_i$ for all $i = 1 \ldots, t$ and define the balls $B_i := \{Q_i \leq s\}$. Then provided that the intersection $\bigcap_{i=1}^t B_i$ has a nonempty interior, the ball $B := \{Q \leq s\}$ is the minimal enclosing ball of the intersection $\bigcap_{i=1}^t B_i$.

Proof. Let R^2 be the square radius of B and let R_i^2 be the square radius of B_i, for $i = 1, \ldots, t$. Using Proposition 1, we deduce

$$\begin{aligned} R^2 = \frac{2}{\alpha}(s - v) &= \frac{2}{\alpha}\left(s - \max_{\lambda \in \Delta_t} \left\{ \frac{\alpha}{2}\sum_{i=1}^t \lambda_i \left(\frac{\alpha}{2}\|c_i\|^2 + v_i\right) - \frac{\alpha}{2}\left\|\sum_{i=1}^t \lambda_i c_i\right\|^2 \right\}\right) \\ &= \min_{\lambda \in \Delta_t} \left\|\sum_{i=1}^t \lambda_i c_i\right\|^2 - \sum_{i=1}^t \lambda_i \left(\|c_i\|^2 + \frac{2}{\alpha}(v_i - s)\right) \\ &= \min_{\lambda \in \Delta_t} \left\|\sum_{i=1}^t \lambda_i c_i\right\|^2 - \sum_{i=1}^t \lambda_i \left(\|c_i\|^2 - R_i^2\right). \end{aligned}$$

The center of B is $c = \sum_{i=1}^t \lambda_i c_i$ where λ is the minimizer of the expression above. Comparing with [7, Theorem 3.2], we see that B is exactly the minimum radius ball enclosing the intersection $\bigcap_{i=1}^t B_i$. □

3.5 Numerical examples

In this section, we numerically illustrate optimality gap convergence in Algorithm 3, and explore how Algorithm 5, the variant of Algorithm 3 with memory, aids performance. To this end, we focus on minimizing two functions: the regularized logistic loss function

$$L(w) := \frac{1}{N}\sum_{i=1}^N \log\left(1 + e^{-y_i w^T x_i}\right) + \frac{\alpha}{2}\|w\|_2^2,$$

where $x_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$ are labeled training data, and the "world's worst" function for first-order methods:

$$f(x) = \frac{B}{2}\left((1-x_1)^2 + \sum_{i=1}^{n-1}(x_i - x_{i+1})^2 + x_n^2\right) + \frac{1}{2}\sum_{i=1}^{n} x_i^2$$

(see [70, Section 2.1.2 and Section 2.1.4]). For the logistic regression examples, we use the LIBSVM [29] data sets a1a ($N = 1605$, $n = 123$) and colon-cancer ($N = 62$, $n = 2000$).

3.5.1 Optimality gap convergence

From inequality (3.2), we get the well-known optimality gap estimate for strongly convex functions

$$f(x) - f^* \leq \frac{\|\nabla f(x)\|_2^2}{2\alpha}. \tag{3.10}$$

How does this estimate compare with the gaps $g_k := f(x_k^+) - v_k$ generated by Algorithm 3? Obviously the answer depends on the point where we evaluate the gap estimate in (3.10). Nonetheless, we can say that the gaps g_k are tighter than the gaps $G_k := \frac{\|\nabla f(x_k)\|_2^2}{2\alpha}$. Indeed, by the definition of v_k, we trivially have $v_k \geq f(x_k) - G_k$ and thus

$$g_k = f(x_k^+) - v_k \leq f(x_k) - v_k \leq G_k.$$

On a relative scale, the difference between g_k and G_k is striking; see Figure 3.3. Notice that G_k is an optimality gap estimate before averaging, and g_k is an optimality gap estimate after averaging; the plots in Figure 3.3 show that optimal quadratic averaging makes great relative progress per iteration.

In Figure 3.4, we plot g_k, the true gaps $f(x_k^+) - f^*$, and the gap estimate in (3.10) at x_k, x_k^+, and c_k for the "world's worst" function and the logistic loss function. The true gaps are the tightest, albeit unknown at runtime. Surprisingly, the gaps $\frac{\|\nabla f(c_k)\|_2^2}{2\alpha}$ are quite bad: several orders of magnitude larger than g_k. So even though the centers c_k may appear to be the focal points of the algorithm, the points x_k^+ are the ones to monitor in practice. Finally we note that the gaps g_k and $\frac{\|\nabla f(x_k^+)\|_2^2}{2\alpha}$ are comparable, even though g_k does not rely on gradient information at x_k^+.

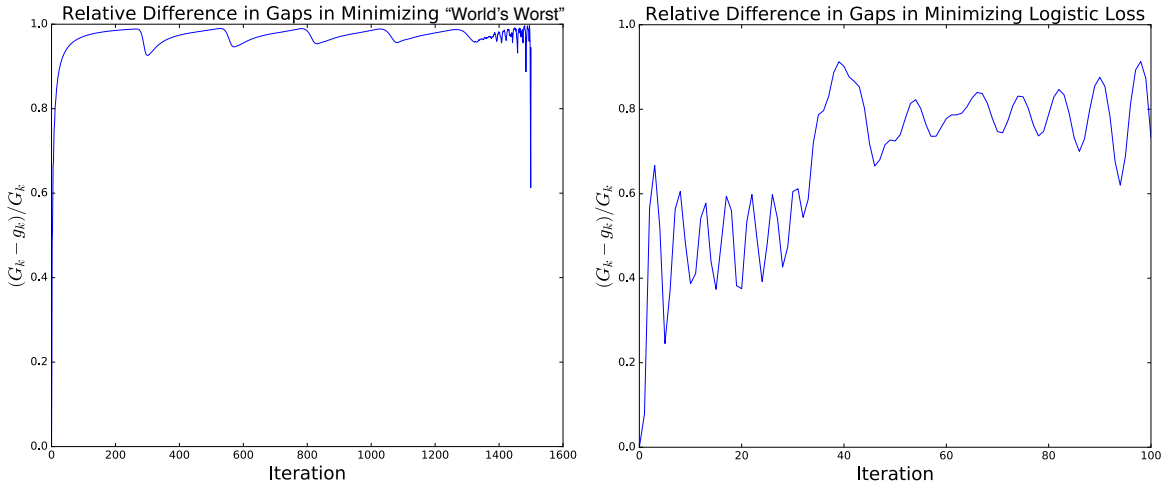


Figure 3.3: Relative differences in gaps $\frac{G_k - g_k}{G_k}$ on the "world's worst" function ($B = 10^6$, $n = 200$), and on the logistic loss on the colon-cancer data set with regularization $\alpha = 0.0001$.

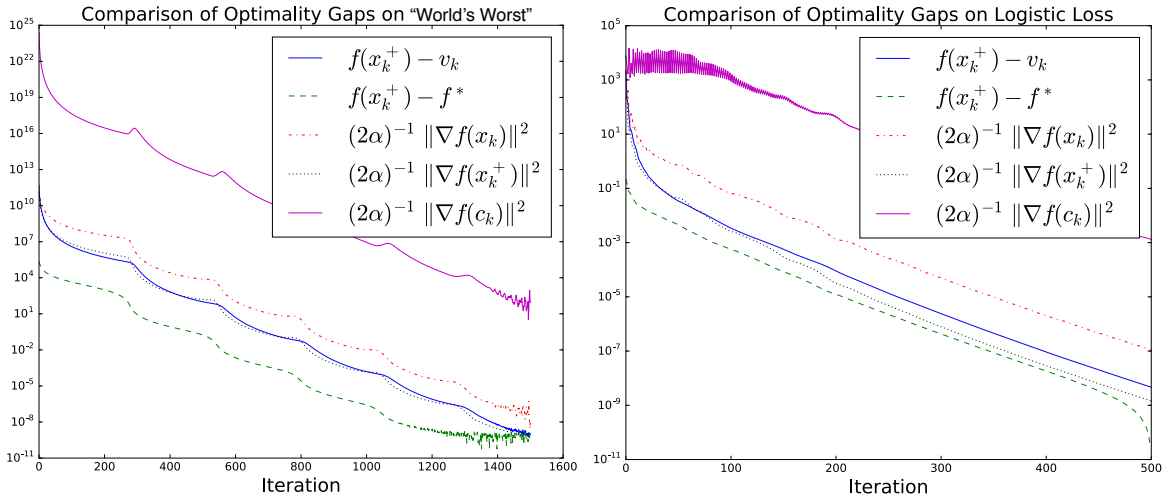


Figure 3.4: Comparison of various optimality gaps on the "world's worst" function ($B = 10^6$, $n = 200$), and on the logistic loss on the a1a data set with regularization $\alpha = 0.0001$.

3.5.2 Optimal quadratic averaging with memory

To demonstrate the effectiveness of optimal quadratic averaging with memory, we use it to minimize the logistic loss (see Figure 3.5). The speedup over the memoryless method is significant, even when taking into account the extra work per iteration needed to solve the small dimensional quadratic subproblems. In Figure 3.6, we compare Algorithm 5 with L-BFGS. The two schemes are on par with each other, and neither is better than the other in all cases.

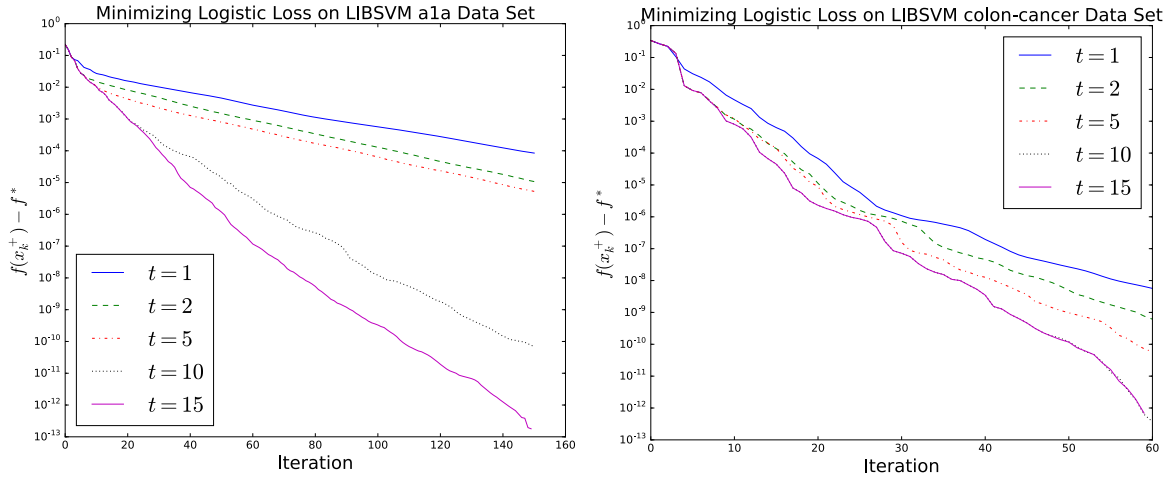


Figure 3.5: Algorithm 5 with various memory sizes t. The case $t = 1$ corresponds to the memoryless optimal averaging method in Algorithm 3. The task is logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer.

It is perhaps fairer to compare L-BFGS with memory size m to Algorithm 5 with memory size $t = 2m$ (see Figure 3.7). Indeed, L-BFGS with memory size m actually stores m *pairs* of vectors, whereas Algorithm 5 with memory size t only stores t vectors. Moreover, the most expensive operation per iteration in L-BFGS requires $4mn$ multiplications (see [73, Algorithm 7.4]); in contrast, computing a new center in Algorithm 5 requires $2n(t + 1)$ multiplications plus the cost of solving a small quadratic program. (Updating the matrix $C^T C$ takes $t + 1$

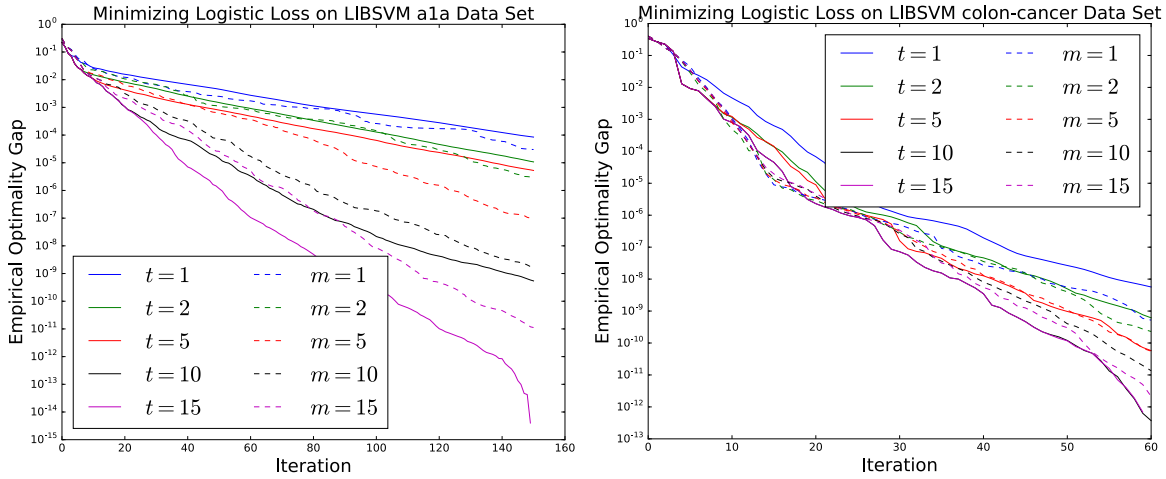


Figure 3.6: Algorithm 5 with memory size t versus L-BFGS with memory size m. The task is logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer.

inner products in $\mathbb{R}^n$, finding λ amounts to solving a small quadratic program, and computing $C\lambda$ takes n inner products in $\mathbb{R}^{t+1}$.) In Figure 3.8, we again compare L-BFGS and Algorithm 5 on logisitic regression, but with less regularization.

We noticed that the small dimensional quadratic program in Algorithm 5 must be solved to high accuracy, especially on poorly conditioned problems; an active-set method works well. Accuracy in the line search is less important. Minimizing the one-dimensional function $r \mapsto f(x+rd)$, with $\|d\| = 1$, to within 10^{-4} accuracy in r works well in general. In Figure 3.9, we show how line search accuracy affects Algorithm 3.

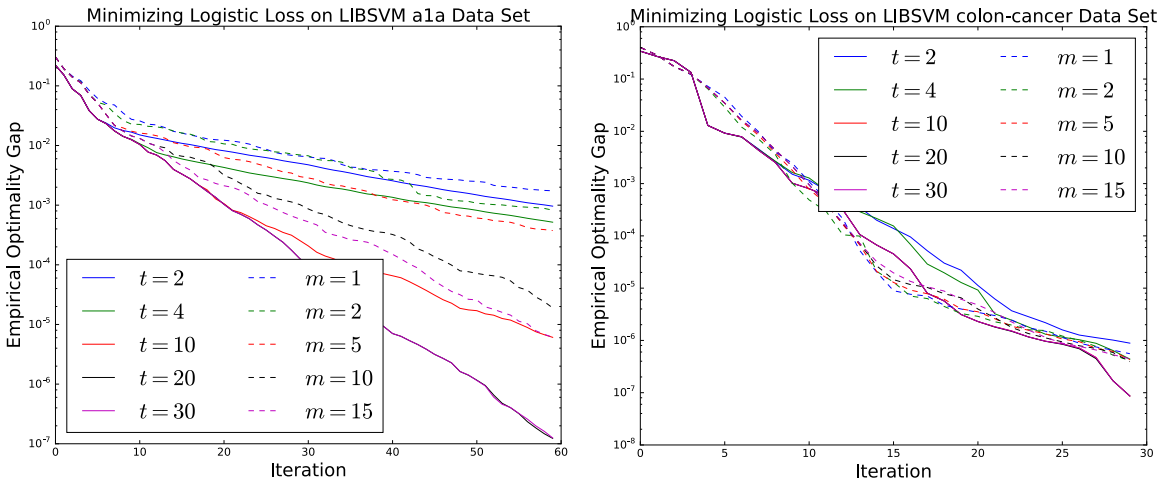


Figure 3.7: A fairer (equal memory) comparison of Algorithm 5 and L-BFGS. The task is still logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer. We focus on lower accuracy than we did in Figure 3.6.

3.6 Comments on proximal extensions

It is natural to try to extend geometric descent and optimal quadratic averaging to a proximal setting. For the sake of concreteness, let us focus on geometric descent. We can easily extend the suboptimal version of the algorithm to the proximal setting, but some difficulties arise when accelerating the method. Suppose we are interested in solving the problem

$$\min_x \ f(x) := g(x) + h(x),$$

where $g\colon \mathbb{R}^n \to \mathbb{R}$ is β-smooth and α-strongly convex, and $h\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is closed, convex, and is such that the proximal mapping

$$\text{prox}_{th}(x) := \operatorname*{argmin}_z \{h(z) + \frac{1}{2t}\|z - x\|^2\}$$

is easily computable. In the analysis of first-order methods for such problems, the *gradient mapping* $G_t(x) := \frac{1}{t}\left(x - \text{prox}_{th}(x - t\nabla g(x))\right)$ plays the role of the usual gradient. The

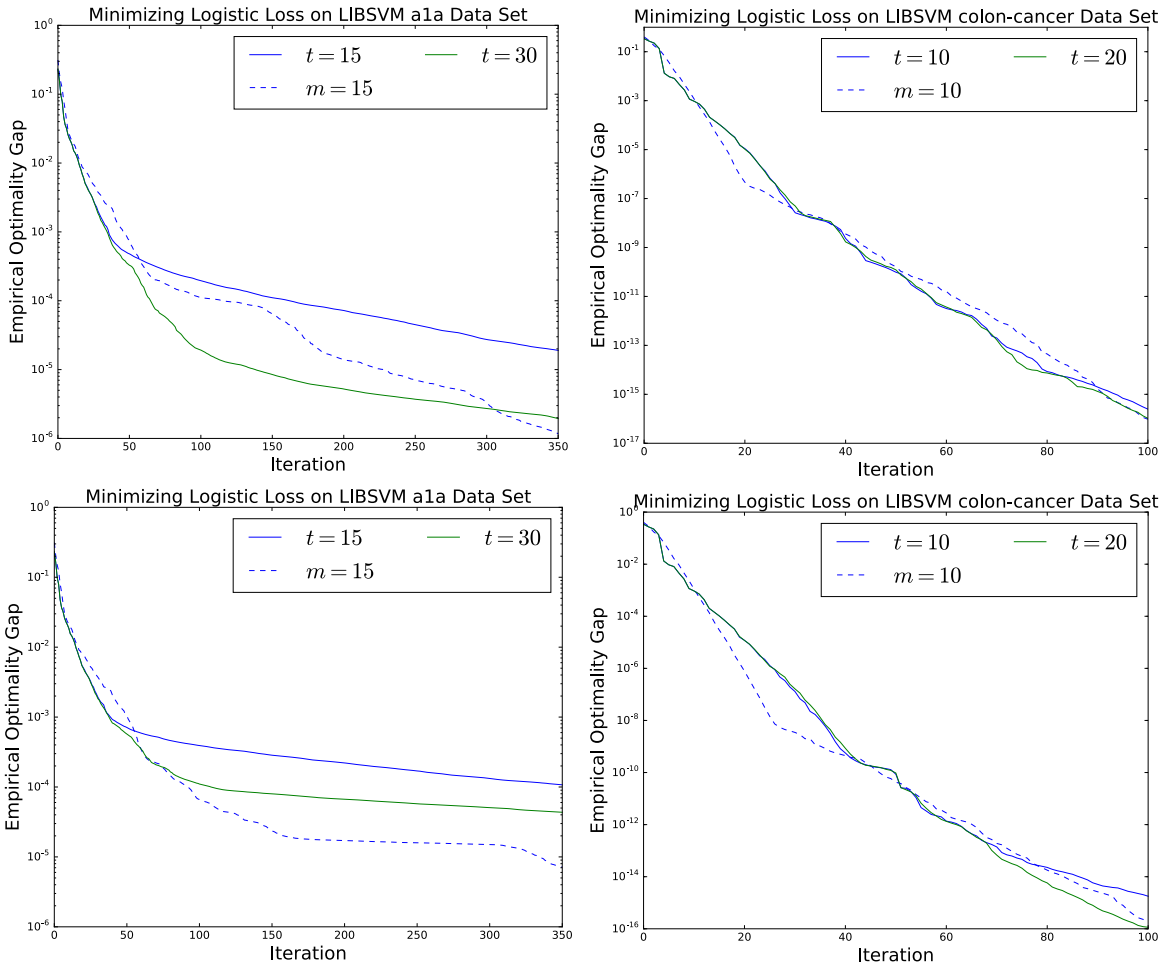


Figure 3.8: Algorithm 5 with memory size t versus L-BFGS with memory size m. The task is logistic regression on data sets a1a and colon-cancer, with $\alpha = 10^{-6}$ (top row) and $\alpha = 10^{-8}$ (bottom row).

following is a standard estimate; see for example [70, Section 2.2.3]. We provide a proof for completeness.

Lemma 6. *Fix a step length $t > 0$ and define a proximal gradient step $x^+ := x - tG_t(x)$.*

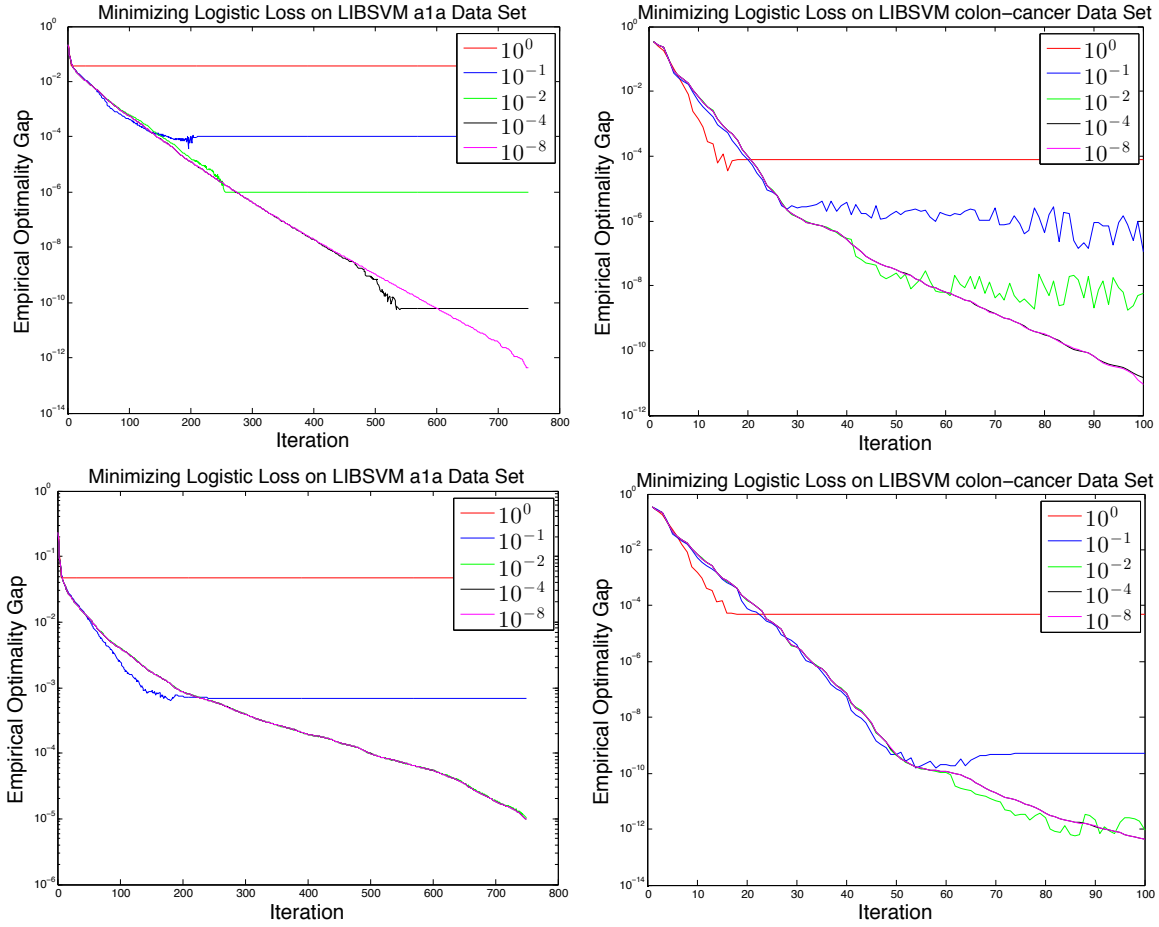


Figure 3.9: A comparison of how the line search tolerance in Algorithm 3 affects convergence. In the top row, we do the comparison with logistic regression on the a1a and colon-cancer data sets with regularization $\alpha = 10^{-4}$. In the bottom row, we use regularization 10^{-8}.

Then for every $y \in \mathbb{R}^n$ the inequality holds:

$$f(y) \geq f(x^+) + \langle G_t(x),\, y - x\rangle + t\left(1 - \frac{\beta t}{2}\right)\|G_t(x)\|_2^2 + \frac{\alpha}{2}\|y - x\|_2^2.$$

Proof. Appealing to β-smoothness of g, we deduce

$$f(x^+) \leq g(x) - t\,\langle \nabla g(x),\, G_t(x)\rangle + \frac{\beta t^2}{2}\|G_t(x)\|_2^2 + h(x^+).$$

Furthermore, strong convexity of g implies

$$f(x^+) \leq g(y) + \left\langle \nabla g(x),\, x^+ - y \right\rangle - \frac{\alpha}{2}\left\|y - x\right\|_2^2 + \frac{\beta t^2}{2}\left\|G_t(x)\right\|_2^2 + h(x^+).$$

Finally, using the observation that $G_t(x) - \nabla g(x)$ belongs to $\partial h(x^+)$, we have

$$f(x^+) \leq f(y) + \left\langle G_t(x),\, x^+ - y \right\rangle - \frac{\alpha}{2}\left\|y - x\right\|_2^2 + \frac{\beta t^2}{2}\left\|G_t(x)\right\|_2^2.$$

Rearrangement completes the proof. □

If we let $y = x^*$ in Lemma 6 and rearrange we get

$$x^* \in B\left(x - \frac{1}{\alpha}G_t(x), \left(\frac{1}{\alpha^2} - \frac{2}{\alpha}t + \frac{\beta}{\alpha}t^2\right)\left\|G_t(x)\right\|_2^2 - \frac{2}{\alpha}\left(f(x^+) - f^*\right)\right).$$

How should we choose the step length t? A simple approach is to choose t to minimize the quantity $\frac{1}{\alpha^2} - \frac{2}{\alpha}t + \frac{\beta}{\alpha}t^2$, i.e., set $t = \frac{1}{\beta}$. With this choice of t, we deduce the inclusion

$$x^* \in B\left(x^{++}, \left(1 - \frac{1}{\kappa}\right)\frac{\left\|G_{1/\beta}(x)\right\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x^+) - f^*\right)\right),$$

where $x^{++} = x - \frac{1}{\alpha}G_{1/\beta}(x)$ is a *long step* and $x^+ = x - \frac{1}{\beta}G_{1/\beta}(x)$ is a *short step*. A proximal version of the suboptimal geometric descent follows easily from Lemma 4.

To accelerate the proximal geometric descent algorithm we assume in iteration k that x^* lies in some ball

$$B\left(c_k, R_k^2 - \frac{2}{\alpha}\left(f(y_k) - f^*\right)\right).$$

We then consider a second minimizer enclosing ball derived from information at some point x_{k+1}:

$$x^* \in B\left(x_{k+1}^{++}, \left(1 - \frac{1}{\kappa}\right)\frac{\left\|G_{1/\beta}(x_{k+1})\right\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right).$$

Following the same pattern as in Section 3.4.2, if we choose x_{k+1} to satisfy $f(x_{k+1}) \leq f(y_k)$ and appeal to the smoothness inequality $f(x_{k+1}^+) \leq f(x_{k+1}) - \frac{1}{2\beta}\left\|G_{1/\beta}(x_{k+1})\right\|_2^2$, we deduce the inclusion

$$x^* \in B\left(c_k, R_k^2 - \frac{1}{\kappa}\frac{\left\|G_{1/\beta}(x_{k+1})\right\|_2^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right).$$

By Lemma 5 there is a new center c_{k+1} with

$$x^* \in B\left(c_{k+1}, \left(1 - \frac{1}{\sqrt{\kappa}}\right) R_k^2 - \frac{2}{\alpha}\left(f(x_{k+1}^+) - f^*\right)\right),$$

provided the old centers x_{k+1}^{++} and c_k are far apart; specifically, we must be sure that the inequality

$$\left\|x_{k+1}^{++} - c_k\right\|_2^2 \geq \frac{\left\|G_{1/\beta}(x_{k+1})\right\|_2^2}{\alpha^2} \qquad \text{holds.}$$

How do we choose x_{k+1} to satisfy both $f(x_{k+1}) \leq f(y_k)$ and $\left\|x_{k+1}^{++} - c_k\right\|_2^2 \geq \frac{\left\|G_{1/\beta}(x_{k+1})\right\|_2^2}{\alpha^2}$? The desired x_{k+1} does exist; for example, $x_{k+1} = x^*$ is such a point. In the proximal setting, it is not clear how to choose x_{k+1} to ensure these two inequalities (even for specific problem classes). This is an interesting topic for future research.

Acknowledgments

We thank the anonymous referee for useful suggestions, which undoubtedly improved the quality of the paper. We also thank Stephen J. Wright for pointing out an important typo in the proof of Theorem 5 in an early version of the manuscript.

3.7 Exact line search in accelerated gradient descent

Nesterov's method is based on an *estimate sequence*; that is, a sequence of functions Q_k and nonnegative numbers Λ_k with

$$\Lambda_k \to 0 \quad \text{and} \quad Q_k(x) \leq (1 - \Lambda_k) f(x) + \Lambda_k Q_0(x).$$

Estimate sequences are useful because if y_k satisfies $f(y_k) \leq v_k := \min_{x \in \mathbb{R}^n} Q_k(x)$, then

$$f(y_k) - f^* \leq \Lambda_k \left(Q_0(x^*) - f^*\right);$$

that is, $f(y_k)$ approaches f^* with error proportional to Λ_k, see [70].

The quadratics in Algorithm 4 (with appropriately chosen Λ_k) form an estimate sequence. To explain, for $k \geq 1$, pick vectors x_k and numbers $\lambda_k \in (\delta, 1)$ with $\delta > 0$. Next, recursively

define

$$Q_0(x) = v_0 + \frac{\gamma_0}{2}\|x - c_0\|_2^2 \quad \text{and}$$
$$Q_k(x) = (1-\lambda_k)Q_{k-1}(x) + \lambda_k\left(f(x_k) - \frac{\|\nabla f(x_k)\|_2^2}{2\alpha} + \frac{\alpha}{2}\left\|x - x_k^{++}\right\|_2^2\right).$$

Then the quadratics Q_k and numbers $\Lambda_k = \prod_{j=1}^{k}(1-\lambda_j)$ are an estimate sequence for f. Nesterov's method is designed to ensure the inequality $f(x_k^+) \leq v_k$ with the added *optimal rate condition* $\lambda_k \geq \sqrt{\frac{\alpha}{\beta}}$.

The scheme in Algorithm 4 with $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$ also guarantees these conditions. Trivially we have $f(x_0^+) \leq v_0$. Assume, for induction, that we have $f(x_{k-1}^+) \leq v_{k-1}$. From [70, Lemma 2.2.3], we know

$$v_k = (1-\lambda_k)v_{k-1} + \lambda_k f(x_k) - \frac{\lambda_k^2}{2\gamma_k}\|\nabla f(x_k)\|_2^2 + \\ + \frac{\lambda_k(1-\lambda_k)\gamma_{k-1}}{\gamma_k}\left(\frac{\alpha}{2}\|x_k - c_{k-1}\|_2^2 + \langle \nabla f(x_k),\, c_{k-1} - x_k\rangle\right).$$

Since $x_k = \texttt{line\_search}\left(c_{k-1}, x_{k-1}^+\right)$, we have $f(x_k) \leq f(x_{k-1}^+) \leq v_{k-1}$ and $\langle \nabla f(x_k),\, c_{k-1} - x_k\rangle = 0$, and therefore

$$v_k \geq f(x_k) - \frac{\lambda_k^2}{2\gamma_k}\|\nabla f(x_k)\|_2^2 = f(x_k) - \frac{1}{2\beta}\|\nabla f(x_k)\|_2^2 \geq f(x_k^+).$$

Provided we set $\gamma_0 \geq \alpha$, we get the optimal rate condition $\lambda_k = \sqrt{\frac{\gamma_k}{\beta}} \geq \sqrt{\frac{\alpha}{\beta}}$.

Chapter 4

LEVEL-SET METHODS FOR CONVEX OPTIMIZATION

Authors

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

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 and constraint functions, and instead approximately solves a sequence of parametric level-set problems. A zero-finding procedure, based on inexact function evaluations and possibly inexact derivative information, leads to an efficient solution scheme 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 generalized linear models for statistical inference.

4.1 Introduction

To motivate the discussion, consider the typical problem of recovering a sparse 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 problem formulations.

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

Computationally, BP_σ is perceived to be the most challenging of the three because of the complicated geometry of the feasible region. For example, a projected- or proximal-gradient method for LS_τ or QP_λ requires relatively little cost per iteration[1] beyond forming the product Ax or $A^T y$. In contrast, a comparable first-order method for BP_σ, such as the alternating direction method of multipliers (ADMM) [18, 41], requires at each iteration the solution of a linear least-squares problem [12] and maintains iterates that are both infeasible and suboptimal. Consequently, problems LS_τ and QP_λ are most often solved in practice, and most algorithm development and implementation targets these versions of the problem. Nevertheless, the formulation BP_σ is often more natural, since the parameter σ plays an entirely transparent role, signifying an acceptable tolerance on the data misfit.

This paper targets optimization problems generalizing the formulation BP_σ. Setting the stage, consider the pair of problems

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad \varphi(x) \quad \text{subject to} \quad \rho(Ax-b) \le \sigma, \tag{$\mathcal{P}_\sigma$}$$

and

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad \rho(Ax-b) \quad \text{subject to} \quad \varphi(x) \le \tau, \tag{$\mathcal{Q}_\tau$}$$

[1] Projection onto the ball $\{x : \|x\|_1 \le \tau\}$ requires $\mathcal{O}(n \log n)$ operations; the proximal map for the function $\lambda\|x\|_1$ requires $\mathcal{O}(n)$ operations.

where $\mathcal{X}$ is a closed convex set, φ and ρ are (possibly infinite-valued) closed convex functions, and A is a linear map. Here, $\mathcal{P}_\sigma$ and $\mathcal{Q}_\tau$ extend the problems BP_σ and LS_τ, respectively. Such formulations are ubiquitous in contemporary optimization and its applications. Our working assumption is that the level-set problem $\mathcal{Q}_\tau$ is easier to solve than $\mathcal{P}_\sigma$—perhaps because it allows for a specialized algorithm for its solution. In §4.4, we discuss a range of problems, including nonsmooth regularization, conic optimization, and generalized linear models, with this property.

Our main goal is to develop a practical and theoretically sound algorithmic framework that can be used 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 to avoid any one formulation based on computational considerations alone. This observation is very significant in applications since, although the formulations $\mathcal{P}_\sigma$ and $\mathcal{Q}_\tau$ as well as their Lagrangian (or penalty) formulation are, in a sense, mathematically and computationally equivalent, they are far from equivalent from a modeling perspective. To illustrate this point, consider a scenario where we wish to compare the performance of various regularizers φ_j, $j = 1, \ldots, k$, for a range of values of the model misfit $\rho(Ax - b) \leq \sigma_i$, $i = 1, \ldots, p$. This is an important task in machine learning applications where one wishes to build a classifier based on training data. In this scenario, the model formulation $\mathcal{P}_\sigma$ is the only one that allows an apples-to-apples comparison between regularizers φ_i for a fixed level of model misfit. We illustrate this point in §4.4.3 on a regularized logistic regression problem.

4.1.1 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*:

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

where OPT is the optimal value of $\mathcal{P}_\sigma$. This terminology is used by Harchaoui, Juditsky, and Nemirovski [42], and we adopt it here. The proposed strategy is based on exchanging the roles of the objective and constraint functions in $\mathcal{P}_\sigma$, and approximately solving a sequence of level-set problems $\mathcal{Q}_\tau$ for varying parameters τ.

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

$$v(\tau) := \min_{x \in \mathcal{X}} \{\rho(Ax - b) \mid \varphi(x) \leq \tau\} \ . \tag{4.1}$$

The univariate function v thus defined is nonincreasing and convex [78, Theorem 5.3]. Under the mild assumption that the constraint $\rho(Ax - b) \leq \sigma$ is active at any optimal solution of $\mathcal{P}_\sigma$, it is easy to see that the value $\tau_* := \text{OPT}$ satisfies the equation

$$v(\tau) = \sigma. \tag{4.2}$$

Conversely, it is immediate that for any $\tau \leq \tau_*$ satisfying $v(\tau) \leq \sigma + \epsilon$, solutions of $\mathcal{Q}_\tau$ are super-optimal and ϵ-feasible for $\mathcal{P}_\sigma$, as required. In summary, we have translated the problem $\mathcal{P}_\sigma$ to that of finding the minimal root of the nonlinear univariate equation (4.2). We show in §4.2 how approximate solutions of $\mathcal{Q}_\tau$ can serve as the basis of a root-finding procedure for this key equation. For more details about the relationship between $\mathcal{P}_\sigma$, $\mathcal{Q}_\tau$, and their value functions, see Aravkin, Burke, and Friedlander [3, Theorem 2.1].

Our technical assumptions on the problem $\mathcal{P}_\sigma$ are relatively few, and so in principle the approach applies to a wide class of convex optimization problems. In order to make this scheme practical, however, it is essential that approximate solutions of $\mathcal{Q}_\tau$ can be efficiently computed over a sequence of parameters τ. Hence, efficient implementations attempt to warm start each new problem. It is thus desirable that the sequence of parameters τ_k increases monotonically, since this guarantees that the approximate solutions of $\mathcal{Q}_{\tau_k}$ are feasible for the next problem in the sequence. Bisection methods do not have this property, and we therefore propose variants of secant and Newton methods that accommodate inexact oracles for v and exhibit the desired monotonicity property. We prove that the resulting root-finding procedures

unconditionally have a global linear rate of convergence. 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).

The outline of the manuscript is as follows. In §4.2, we prove 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; see Example 4.4.3. In §4.3, we consider refinements to the overall method, focusing on linear least-squares constraints and recovering feasibility. Section 4.4 explores level-set methods in notable optimization domains, including semi-definite programming, gauge optimization, regularized regression, and generalized linear models. We also describe the specific steps needed to implement the root-finding approach for some representative applications, including low-rank matrix completion [60, 75], sensor-network localization [13, 14, 15], and group detection via the elastic net [93].

Related work

The intuition behind the proposed framework has a distinguished history, appearing even in antiquity. Perhaps the earliest instance is Queen Dido's problem and the fabled origins of Carthage [37, Page 548]. In short, the 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 provides the foundation for the Markowitz mean-variance portfolio theory [62]; the basic problem is to choose a portfolio of financial instruments having a lower-bounded rate of return that minimizes the volatility (variance) of the portfolio. The converse problem is to maximize the rate of return with a bound on volatility. Numerous other examples occur throughout history, and the great variety of

possible modern applications is formalized by the inverse function theorem in Aravkin et al. [3, Theorem 2.1]. More generally, the underlying idea of the trade-offs between various objectives form the foundations for multi-objective optimization [65].

In the context of numerical optimization, our work is motivated by the widely-used SPGL1 algorithm [85, 86] for the 1-norm regularized least-squares problem and its extensions [3]. A shortcoming of the numerical theory to date is the absence of practical complexity and convergence guarantees. In this work, we (*i*) take a fresh new look at this general framework, (*ii*) provide rigorous convergence guarantees, (*iii*) further illustrate the vast applicability of the approach, and (*iv*) show how the proposed framework can be instantiated in concrete circumstances.

Related ideas appear in Lemaréchal, Nemirovskii, and Nesterov [55], who develop their *level* and *truncated level* methods using bundle ideas for convex optimization [54, 89]. Their algorithm is similar in spirit since they work with lower-level sets of the objective function. They consider the convex optimization problem

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

where each function f_j is convex and $\mathcal{X}$ is a nonempty closed convex set. The authors define the function

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

Their algorithm constructs the smallest solution τ_* to the equation $g(\tau) = 0$; then τ_* is the optimal value of the original convex program. See also Nesterov [70, §3.3.4] for a discussion.

More recently, Harchoui et al. [42], in a paper inspired by Lemaréchal et al. [55], present an algorithm focusing on instances of the problem $\mathcal{P}_\sigma$, where ρ is smooth and φ is a gauge of 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 for generating lower bounds and affine minorants on the value function. In contrast, our root finding phase is agnostic to the inner evaluation algorithm, as is the case in the approaches described by Aravkin et al. [3] and van den Berg and Friedlander [85, 86]. Consequently, we see that affine minorants are naturally

obtained from dual certificates in full generality. This is in particular the case for the affine minorants derived from the Frank-Wolfe algorithm; see §4.2.3. This observation immediately opens the door to the use of other primal-dual algorithms, and more generally, to algorithms for solving the primal and dual problems in parallel.

4.1.2 Notation

The notation we use is standard, and follows closely that in Rockafellar's monograph [78]. The functions we consider take values in the extended real line $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$. For any function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$, we use the symbol $[f \leq \alpha] := \{x \in \mathbb{R}^n\colon f(x) \leq \alpha\}$ to denote the α-sublevel set. The *domain* and the *epigraph* of f are defined by

$$\operatorname{dom} f := \{x \in \mathbb{R}^n\colon f(x) < +\infty\} \quad \text{and} \quad \operatorname{epi} f := \{(x, r) \in \mathbb{R}^n \times \mathbb{R}\colon r \geq f(x)\},$$

respectively. We say that f is closed if its epigraph $\operatorname{epi} f$ is a closed set. An *affine minorant* of f is any affine function g satisfying $g(x) \leq f(x)$ for all x. The *subdifferential* of a convex function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ at a point $x \in \operatorname{dom} f$ is the set

$$\partial f(x) := \{v \in \mathbb{R}^n \mid f(y) \geq f(x) + \langle v,\, y - x\rangle \text{ for all } y \in \mathbb{R}^n\}.$$

The *Fenchel conjugate* of f is the closed, convex function $f^\star\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ defined by

$$f^\star(y) := \sup_x \{\langle x,\, y\rangle - f(x)\}.$$

The subdifferential and the conjugate of a convex function f are related by the *Fenchel-Young inequality*: any two points x and y satisfy the inequality

$$f(x) + f^\star(y) \geq \langle y, x\rangle.$$

Moreover, equality holds if and only if $y \in \partial f(x)$. For any set $\mathcal{C}$ in $\mathbb{R}^n$, we define the associated indicator function

$$\delta_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{otherwise.} \end{cases}$$

The conjugate of the indicator function is simply the support function $\delta_{\mathcal{C}}^{\star}(y) = \sup_{x \in \mathcal{C}} \langle x,\, y \rangle$. In particular, for any norm $\|\cdot\|$, the support function of the unit ball $\{x\colon\ \|x\| \leq 1\}$ is the dual norm. The p-norms and corresponding closed unit balls are denoted by $\|\cdot\|_p$ and $\mathbb{B}_p$, respectively. For any convex cone $\mathcal{K}$, the *dual cone* is defined by

$$\mathcal{K}^* := \{y \mid \langle x, y \rangle \geq 0 \text{ for all } x \in \mathcal{K}\}.$$

We always endow the Euclidean space of real $m \times n$ matrices $\mathbb{R}^{m \times n}$ with the trace product $\langle X, Y \rangle := \operatorname{tr}(X^T Y)$ and the induced Frobenius norm $\|X\|_F := \sqrt{\langle X, X \rangle}$. For any matrix $X \in \mathbb{R}^{m \times n}$, the symbols $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_{\min\{m,n\}}(X)$ denote the singular values of X. 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. For any symmetric matrix $X \in \mathcal{S}^n$, the symbols $\lambda_1(X) \geq \lambda_2(X) \geq \cdots \geq \lambda_n(X)$ denote the eigenvalues of X. The closed, convex cone of $n \times n$ positive semi-definite matrices is denoted by $\mathcal{S}_+^n = \{X \in \mathcal{S}^n : X \succeq 0\}$. Both the nonnegative orthant $\mathbb{R}_+^n$ and the positive semi-definite cone $\mathcal{S}_+^n$ are self-dual. The symbol $e \in \mathbb{R}^n$ denotes the vector of all ones.

4.2 Root-finding with inexact oracles

Approximate solutions of $\mathcal{Q}_\tau$ are central to our algorithmic framework, since this is the oracle through which we access v. The available algorithms for $\mathcal{Q}_\tau$ dictate the quality of the oracle. 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. The algorithms presented here apply to any convex nonincreasing 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 $0 \leq f(\tau) \leq \epsilon$.

4.2.1 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 3 (Inexact evaluation oracle)**.** *For a function $f : \mathbb{R}_+ \to \mathbb{R}$, an* inexact evaluation oracle *is a map $\mathcal{O}_f$ that assigns to each pair $(\tau, \alpha) \in [f > 0] \times [1, \infty)$ real numbers (ℓ, u) such that $0 < \ell \leq f(\tau) \leq u$ and $u/\ell \leq \alpha$.*

Note that this oracle guarantees a relative accuracy $u/\ell \leq \alpha$, rather than one based on the absolute gap $u - \ell$. This allows the oracle to be increasingly inexact (and presumably cheaper) for larger values of $f(\tau)$. The relative-accuracy condition is no less general than one based on an absolute gap. In particular, it is readily verified that for any numbers l, u that satisfy $0 \leq \ell \leq f(\tau) \leq u$ and $u - \ell \leq (1 - 1/\alpha)\epsilon$, either

- τ is an ϵ-approximate root, i.e., $f(\tau) \leq \epsilon$; or
- the relative-accuracy condition $1 \leq u/\ell \leq \alpha$ is valid.

Indeed, provided $f(\tau) > \epsilon$, we deduce $u/\ell \leq 1 + (1 - 1/\alpha)\epsilon/\ell \leq 1 + (1 - 1/\alpha)u/\ell$, which after rearranging terms yields the desired inequality $u/\ell \leq \alpha$. Hence, the cost of evaluating $f(\tau)$ within an additive error directly translates into a cost of the same order for evaluating $f(\tau)$ up to relative accuracy. Algorithm 7 outlines a secant method based on the inexact evaluation oracle. Theorem 4.2.1 establishes the corresponding global convergence guarantees; the proof appears in Appendix 4.5.

Theorem 4.2.1 (Linear convergence of the inexact secant method)**.** *The inexact secant method (Algorithm 7) 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 $s_1 := (u_0 - \ell_1)/(\tau_0 - \tau_1)$.

Algorithm 7: Inexact secant method

Data: A decreasing convex function $f : \mathbb{R}_+ \to \mathbb{R}$ via an inexact evaluation oracle $\mathcal{O}_f$; target accuracy $\epsilon > 0$; initial points τ_0, τ_1 with $0 \leq \tau_0 < \tau_1$ such that $f(\tau_0) \geq f(\tau_1) > 0$; constant $\alpha \in (1, 2)$.

$(\ell_0, u_0) \leftarrow \mathcal{O}_f(\tau_0, \alpha)$

$k \leftarrow 1$

while $u_k > \epsilon$ **do**

- $(\ell_k, u_k) \leftarrow \mathcal{O}_f(\tau_k, \alpha)$ `[oracle evaluation for lower/upper bounds]`
- $u_k \leftarrow \min\{u_k, u_{k-1}\}$ `[ensure upper bound decreases]`
- $s_k \leftarrow (u_{k-1} - \ell_k)/(\tau_{k-1} - \tau_k)$ `[slope of linear approximation]`
- $\tau_{k+1} \leftarrow \tau_k - \ell_k/s_k$ `[secant iteration]`
- $k \leftarrow k + 1$

end

return τ_k

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 4.1 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 v to find a root of (4.2), the algorithm's indifference to degeneracy translates to an insensitivity to the "width" [76] of the feasible region of $\mathcal{P}_\sigma$—an unsurprising consequence of the fact that the scheme maintains infeasible iterates for $\mathcal{P}_\sigma$. Thus such methods are well-suited for problems $\mathcal{P}_\sigma$ for which the Slater constraint qualification is close to failing. On the other hand, for non-degenerate problems, we can hope for superlinear convergence when the function is evaluated with sufficient accuracy (see

Theorem 4.5.1).

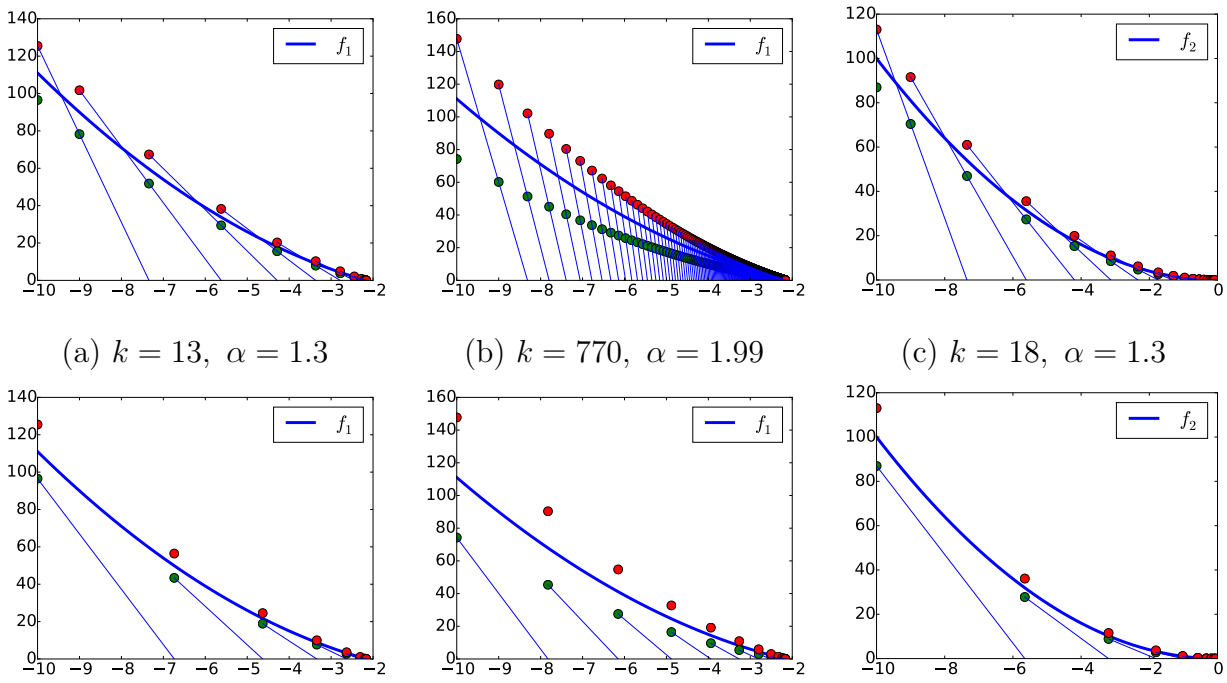
Observe that the iteration bound in Theorem 4.2.1 is infinite for $\alpha \geq 2$. Surprisingly, this is not an artifact of the proof. As illustrated by Figure 4.1(b), the inexact secant method behaves poorly for α close to 2. Indeed, it can fail to converge linearly (or at all) to the minimal root for any $\alpha \geq 2$, 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 $\alpha \geq 2$, the quotients q_k tend to one, indicating that the method stalls.

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

Definition 4 (Affine minorant oracle)**.** *For a function $f : \mathbb{R}_+ \to \mathbb{R}$, an* affine minorant oracle *is a mapping $\mathcal{O}_f$ that assigns to each pair $(t, \alpha) \in [f > 0] \times [1, \infty)$ real numbers (ℓ, u, s) such that $0 < \ell \leq f(\tau) \leq u$ and $u/\ell \leq \alpha$, and the affine function $\tau' \mapsto \ell + s(\tau' - \tau)$ globally minorizes f.*

Algorithm 8 outlines a Newton method based on the affine minorant oracle. The inexact Newton method enjoys global convergence guarantees analogous to those of the inexact secant method, as described by Theorem 4.2.2; see Appendix 4.5 for the proof.



(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) $k = 10,\ \alpha = 1.3$

Figure 4.1: 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 $f_i(\tau)$.

Theorem 4.2.2 (Linear convergence of the inexact Newton method)**.** *The inexact Newton method (Algorithm 8) terminates after at most*

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

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 $(\ell_k, u_k, s_k) =$

Algorithm 8: Inexact Newton method

Data: Convex decreasing function $f\colon \mathbb{R}_+ \to \mathbb{R}$ via an affine minorant oracle $\mathcal{O}_f$; target accuracy $\epsilon > 0$; initial point τ_0 with $f(\tau_0) > 0$; constant $\alpha \in (1, 2)$.

$u_{-1} \leftarrow +\infty$

$k \leftarrow 0$

while $u_k > \epsilon$ **do**

- $(\ell_k, u_k, s_k) \leftarrow \mathcal{O}_f(\tau_k, \alpha)$ `[evaluate lower affine minorant oracle]`
- $u_k \leftarrow \min\{u_k, u_{k-1}\}$ `[ensure upper bound decreases]`
- $\tau_{k+1} \leftarrow \tau_k - \ell_k/s_k$ `[Newton iteration]`
- $k \leftarrow k + 1$

end

return τ_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 $-\ell_k/s_k^{\text{secant}}$.

As might be expected, the Newton method often outperforms the secant method in practice. The bottom row of panels in Figure 4.1 shows the progress of the Newton method on the same degenerate and nondegenerate test problems discussed earlier. Note in particular that 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 with 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.

4.2.3 Lower minorants from duality

Under what circumstances are affine minorant oracles of the value function v readily available? Not surprisingly, duality delivers an answer. Suppose we can express the value function in dual form

$$v(\tau) = \max_{y} \ \Phi(y, \tau),$$

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

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

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

$$\Phi(y, \tau) := \langle b, \, y \rangle - \rho^\star(-y) - \delta^\star_{\mathcal{X} \cap [\varphi \leq \tau]}(A^T y) \tag{4.3}$$

yields an explicit representation for Φ. Note that convexity of Φ in τ is immediate; see Lemma 4.5.2.

Many standard first-order methods that might be used as an oracle for evaluating $v(\bar{\tau}) - \sigma$, 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 [66], Frank-Wolfe [39, 45], some projected (sub)gradient methods [6], and accelerated versions [68, 82, 83]. Whenever such a dual certificate $\bar{y}$ is available, we have

$$\begin{aligned} v(\tau) - \sigma \geq \Phi(\bar{y}, \tau) - \sigma &= \big(\Phi(\bar{y}, \bar{\tau}) - \sigma\big) + \big(\Phi(\bar{y}, \tau) - \Phi(\bar{y}, \bar{\tau})\big) \\ &\geq \bar{\ell} + \bar{s}(\tau - \bar{\tau}), \end{aligned} \tag{4.4}$$

where $\bar{s}$ is any subgradient of Φ at $(\bar{y}, \bar{\tau})$ with respect to τ. Hence, an inexact evaluation oracle that uses dual certificates can always be upgraded to an affine minorant oracle provided that an element of the subdifferential $\partial_\tau \Phi(y, \tau)$ can be evaluated. In the context of (4.3), this amounts to being able to compute an element of $\partial_\tau \delta^\star_{\mathcal{X} \cap [\varphi \leq \tau]}(A^T y)$. Reassuringly, such

subdifferential formulas are readily available for a huge class of contemporary problems [3, Equations 4.1b, 6.5d, 6.20], and, in particular, for all the problems discussed in the rest of the paper.

In some instances, lower-bounds on the optimal value of $\mathcal{Q}_\tau$ 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. Supposing that the function ρ is smooth, the Frank-Wolfe method applied to the problem $\mathcal{Q}_\tau$ iterates the following two steps:

$$\left\{\begin{array}{rcl} z_k & = & \operatorname*{argmin}_{z \in \mathcal{X} \cap [\varphi \leq \tau]} \langle A^T \nabla \rho(Ax_k - b), z \rangle \\ x_{k+1} & = & x_k + t_k(z_k - x_k) \end{array}\right. \tag{4.5}$$

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

$$u_k = \min_{i=1,\ldots,k} \rho(Ax_i - b)$$

on the optimal value of $\mathcal{Q}_\tau$. Moreover, it is easy to deduce from convexity that the following are valid lower bounds:

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

Jaggi [45] 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, Juditsky, and Nemirovski [42] observe how to deduce from such lower bounds ℓ_k an affine minorant of the value function v, leading to a level-set scheme based on Newton's method.

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

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

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

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

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 $\partial_\tau\delta^\star_{\mathcal{X}\cap[\varphi\le\tau]}(A^T y_k)$.

4.3 Refinements

This section can be considered as an aside in our main exposition. Here, we address 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.

4.3.1 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

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

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

Throughout this section, the problems $\mathcal{P}_\sigma$ and $\mathcal{Q}_\tau$ continue to define the original formulations without the squares.

This straightforward adaptation, however, presents some numerical difficulties. Following the strategy outlined in the previous sections, the root finding procedure for $\mathcal{P}_\sigma^2$ would be automatically applied to the function

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

where v is the value function corresponding to the original (unsquared) level-set problem $\mathcal{Q}_\tau$. Clearly, the function f_2 is degenerate at each of its roots. As a result, the secant and Newton root-finding methods, respectively, would not converge locally superlinearly or quadratically—even if the values $v(\tau)$ are evaluated exactly. 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

$$f_1(\tau) := v(\tau) - \sigma$$

corresponding to the value function of $\mathcal{Q}_\tau$, but solve $\mathcal{Q}_\tau^2$ to approximately evaluate f_2 and consequently to approximately evaluate f_1. The oracle definitions required for the secant (Algorithm 7) and Newton (Algorithm 8) 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 from those of f_2. The required modifications are described in turn 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

$$\mathcal{O}_{f_2}\left(\tau, \alpha^2\right) = \left(\tfrac{1}{2}\ell^2 - \tfrac{1}{2}\sigma^2,\ \tfrac{1}{2}u^2 - \tfrac{1}{2}\sigma^2\right), \tag{4.6}$$

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

$$0 \leq \ell - \sigma \leq f_1(\tau) \leq u - \sigma \qquad \text{and} \qquad \frac{u-\sigma}{\ell-\sigma} \leq \sqrt{\frac{u^2-\sigma^2}{\ell^2-\sigma^2}} \leq \alpha. \tag{4.7}$$

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: the nuance is in 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

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

As described by (4.6), an inexact solution of $\mathcal{Q}_\tau^2$ delivers values ℓ and u that satisfy (4.7). Suppose that the oracle additionally delivers a dual certificate y that satisfies $\Phi_2(y,\tau) = \frac{1}{2}\ell^2$. Let $s \in \partial_\tau \Phi_2(y,\tau)$ be any subgradient. The following result establishes that

$$(\hat{\ell},\, u,\, s/\|y\|_2) \qquad \text{with} \qquad \hat{\ell} := \Phi_1\left(y/\|y\|_2,\, \tau\right),$$

defines a valid affine minorant for f_1.

Proposition 4.3.1. *The inequalities*

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

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

The proof is given in Appendix 4.5. 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

$$\tfrac{1}{2}u^2 - \tfrac{1}{2}\ell^2 \leq \tfrac{1}{2}(1-1/\alpha)^2\epsilon^2.$$

Then we obtain the inequality

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

Thus, by the discussion following Definition 3, either the whole Newton scheme can now terminate with $f_1(\tau) \leq \epsilon$ or we have achieved the relative accuracy $(u-\sigma)/(\ell-\sigma) \leq \alpha$ for the oracle.

4.3.2 Recovering feasibility

A potential shortcoming of the level-set approach is that the computed solutions are only ϵ-feasible. Some applications may demand feasible solutions. A straightforward remedy is to project the computed ϵ-feasible point onto the original constraint set $\{x \in \mathcal{X} \mid \rho(Ax-b) \leq \sigma\}$. However, this operation can be computationally impractical; for example, access to the matrix A is often only available through matrix vector products. An alternative is provided by Renegar [77], who suggests an inexpensive radial-projection scheme for conic optimization that generates a feasible point while still preserving some notion of optimality. The approach requires knowledge of a point e strictly feasible for the original problem, and obtains a feasible point x whose optimality is measured with respect to e, i.e.,

$$\frac{\varphi(x) - \mathrm{OPT}}{\varphi(e) - \mathrm{OPT}} \leq \delta$$

for some small positive parameter δ.

To explain the approach, fix some target $\delta < 1$ and suppose that $e \in \mathcal{X}$ is strictly feasible for $\mathcal{P}_\sigma$, i.e.,

$$\rho(Ae - b) < \sigma.$$

Suppose also that a point $z \in \mathcal{X}$ is super-optimal and ϵ-feasible for $\mathcal{P}_\sigma$:

$$\varphi(z) \le \text{OPT} \quad \text{and} \quad \sigma < \rho(Az - b) \le \sigma + \epsilon, \qquad \text{with} \qquad \epsilon := \delta\big[\sigma - \rho(Ae - b)\big].$$

These relationships imply the inequality

$$\alpha := \frac{\rho(Az - b) - \sigma}{\rho(Az - b) - \rho(Ae - b)} \le \delta.$$

Set $x := z + \alpha(e - z)$, which is the radial projection of z towards the feasible point e. It follows from convexity that $\varphi(x) \le (1 - \alpha)\varphi(z) + \alpha\varphi(e)$. Subtract OPT from both sides and rearrange terms to obtain

$$\frac{\varphi(x) - \text{OPT}}{\varphi(e) - \text{OPT}} \le (1 - \alpha)\frac{\varphi(z) - \text{OPT}}{\varphi(e) - \text{OPT}} + \alpha \le \alpha \le \delta.$$

It only remains to show that the radial projection x is feasible. The inclusion $x \in \mathcal{X}$ follows from convexity of $\mathcal{X}$. Use the definition of α, together with the convexity of ρ, to obtain

$$\rho(Ax - b) \le \rho(Az - b) - \alpha[\rho(Az - b) - \rho(Ae - b)] = \sigma,$$

which establishes feasibility of x.

4.4 Some problem classes

There is a surprising variety of useful problems that can be treated by the root-finding approach. These include problems from sparse optimization, with applications in compressed sensing and sparse recovery, generalized linear models, which feature prominently in statistical applications, and conic optimization, which includes semidefinite programming. The following sections are in some sense a "cookbook" that describes how features of particular problems can be combined to apply the root-finding approach. In some cases, such as with conic optimization, we have the opportunity to derive unexpected algorithms.

4.4.1 Conic optimization

The general conic problem (CP) has the form

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

Problem	$\mathcal{P}_\sigma$	$\mathcal{Q}_\tau$	Dual of $\mathcal{Q}_\tau$
CP	$\min_x \ \langle c, x\rangle$	$\min_x \ \|\mathcal{A}x - b\|_2$	$\max_{y,\ \mu\geq 0} \ \langle b, y\rangle - \mu\tau$
least-	s.t. $\mathcal{A}x = b$	s.t. $\langle c, x\rangle \leq \tau$	s.t. $\|y\|_2 \leq 1$
squares	$x \in \mathcal{K}$	$x \in \mathcal{K}$	$\mu c - \mathcal{A}^* y \in \mathcal{K}^*$
level			
CP	$\min_x \ \langle c, x\rangle$	$\min_x \ -\lambda_{\min}(x)$	$\max_{y,\ \mu\geq 0} \ \langle b, y\rangle - \mu\tau$
cone	s.t. $\mathcal{A}x = b$	s.t. $\mathcal{A}x = b$	s.t. $\langle \mu c - \mathcal{A}^* y, e\rangle = 1$
level	$x \in \mathcal{K}$	$\langle c, x\rangle \leq \tau$	$\mu c - \mathcal{A}^* y \in \mathcal{K}^*$

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

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 [10] 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. The first 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 via a kind of 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 $\rho(Ax - b) \leq \sigma$, but here we restrict our attention to the simpler affine constraint, which conforms to the standard form of conic optimization.

First approach: least-squares level set

To get started with this approach, we make the blanket assumption that we know a *strictly feasible* vector $\widehat{y}$ for 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

$$\underset{x}{\text{minimize}} \quad \|\mathcal{A}x - b\|_2 \quad \text{subject to} \quad \langle \widehat{c},\, x\rangle \leq \tau,\ x \in \mathcal{K}. \tag{4.8}$$

Our aim is then to find a root of (4.2), where v is the value function of (4.8). The top row of Table 4.1, gives the corresponding dual

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

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 (4.8), 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

$$\bar{\mu} := \min_{\mu} \left\{ \mu\widehat{c} - \frac{\mathcal{A}^* b}{\|b\|_2} \in \mathcal{K}^* \right\}. \tag{4.9}$$

Note the inequality $\bar{\mu} > 0$, because otherwise we would deduce $\mathcal{A}^* b \in -\mathcal{K}^*$, implying the inequality $\|b\|_2^2 = \langle b, \mathcal{A}x\rangle = \langle \mathcal{A}^* b, x\rangle \leq 0$ for any feasible x. This 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 $\mathcal{A}^* b/\|b\|_2$; if $\mathcal{K}$ is the semidefinite cone and $\widehat{c} = I$, the number $\bar{\mu}$ is the right-most eigenvalue of $\mathcal{A}^* b/\|b\|_2$. With these values, Theorem 4.2.2 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

$$C = \max\left\{\bar{\mu} \cdot (\mathrm{OPT} - \langle b, \widehat{y}\rangle), \|b\|_2\right\},$$

the point $x \in \mathcal{K}$ that yields the final upper bound in (4.8) is a super-optimal and ϵ-feasible solution of the shifted CP, i.e.,

$$\langle \widehat{c}, x\rangle \leq \mathrm{OPT} - \langle \widehat{y}, b\rangle \qquad \text{and} \qquad \|\mathcal{A}x - b\|_2 \leq \epsilon.$$

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

$$\langle \widehat{c}, x\rangle = \langle c, x\rangle - \langle \mathcal{A}^*\widehat{y}, x\rangle = \langle c, x\rangle - \langle \widehat{y}, \mathcal{A}x - b\rangle - \langle \widehat{y}, b\rangle \geq \langle c, x\rangle - \langle \widehat{y}, b\rangle - \epsilon\|\widehat{y}\|_2,$$

and hence $\langle c, x\rangle \leq \mathrm{OPT} + \epsilon\|\widehat{y}\|_2$. In particular, in the important case where $c \in \operatorname{int}\mathcal{K}^*$, we deduce super-optimality $\langle c, x\rangle \leq \mathrm{OPT}$ for the target problem CP.

Each Newton root-finding iteration requires an approximate solution of (4.8). As described in §4.3.1, we obtain this approximation by instead solving its smooth formulation with the squared objective $\frac{1}{2}\|\mathcal{A}x - b\|_2^2$. Let $L := \|\mathcal{A}\|_2^2$ be the Lipschitz constant for the gradient $\mathcal{A}^T(\mathcal{A}x - b)$, and 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 §4.3.1), which terminates in at most

$$\mathcal{O}\left(\frac{\sqrt{L} \cdot \tau D}{\epsilon(1 - 1/\alpha)}\right) = \mathcal{O}\left(\frac{\|A\|_2 \cdot D \cdot (\mathrm{OPT} - \langle b, \widehat{y}\rangle)}{\epsilon(1 - 1/\alpha)}\right)$$

iterations [11, §6.2]. Here, we have used the monotonicity of the root finding scheme to conclude $\tau \leq \mathrm{OPT} - \langle b, \widehat{y}\rangle$. When $\mathcal{K}$ is the non-negative orthant, each projection can be accomplished with $\mathcal{O}(n)$ floating point operations [20], while for the semidefinite 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 3. We note that 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 [8] or Tseng [83, §3.1]. We leave the details to the reader.

In summary, we can obtain a point $x \in \mathcal{K}$ that satisfies

$$\langle c, x\rangle \leq \text{OPT} + \epsilon\|\widehat{y}\|_2 \qquad \text{and} \qquad \|\mathcal{A}x - b\|_2 \leq \epsilon$$

in at most

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

iterations of an accelerated projected-gradient method, where $\bar{\mu}$ is defined in (4.9). Reassuringly, the complexity bound depends on all the expected quantities.

Second approach: conic level set

Renegar's recent work [77] 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.,

$$\kappa(x) \leq 0 \qquad \text{if and only if} \qquad x \in \mathcal{K}. \tag{4.10}$$

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

$$\underset{x}{\text{minimize}} \quad \langle c,\, x\rangle \quad \text{subject to} \quad \mathcal{A}x = b,\ \kappa(x) \leq 0, \tag{4.11}$$

which allows us to define the level-set problem

$$\underset{x}{\text{minimize}} \quad \kappa(x) \quad \text{subject to} \quad \mathcal{A}x = b,\ \langle c,\, x\rangle \leq \tau. \tag{4.12}$$

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, which explains the notation. As is shown in [77, 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. 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. We note in passing that 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 4.5.3.

In principle, the main requirement of our level-set approach is that the surrogate function that satisfies (4.10) yields the equivalent formulation (4.11). Depending on the algorithms available for solving the level-set problem (4.12), it may be convenient to define a function κ with certain useful properties. 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 (4.8) and (4.12). 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.

4.4.2 Gauge optimization

In this section, we illustrate the general applicability of the level-set approach to regularized data-fitting problems by restricting the convex functions φ and ρ to be *gauges*—i.e., functions that are additionally nonnegative, positively homogeneous, and vanish at the origin. Throughout, we assume that the side constraint $x \in \mathcal{X}$ is absent from the formulation $\mathcal{P}_\sigma$. A large class of problems of this type occurs in sparsity optimization. Basis pursuit (and its "denoising" variant BP_σ) [32] was our very first example in §4.1, and many related problems can be similarly expressed. The first two columns of Table 4.2 describe various formulations of current interest, including basis pursuit denoising (BPDN), low-rank matrix recovery [26, 38], a sharp version of the elastic-net problem [93], and gauge optimization [40] 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.

The dual representation (4.3) can be specialized for this family, and requires some basic facts regarding a gauge function f and its *polar*

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

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 [40, 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 (4.3), 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

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

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

$$\underset{y}{\text{maximize}} \quad \langle b,\, y\rangle - \tau\varphi^\circ(A^T y) \quad \text{subject to} \quad \rho^\circ(-y) \le 1. \tag{4.13}$$

In the last three rows of the table, we set $\rho = \|\cdot\|_2$, which is self polar. For BPDN, we use the vector 1-norm $\varphi = \|\cdot\|_1$, whose polar is the dual norm $\varphi^\circ = \|\cdot\|_\infty$. For matrix completion, the function $\varphi = \|\cdot\|_* := \sum_{i=1}^{\min\{m,n\}} \sigma_i(\cdot)$ is the nuclear norm of a n-by-m matrix, which is polar to the spectral norm $\varphi^\circ = \sigma_{\max}(\cdot)$. For the sharp elastic net, we use Lemma 4.5.4 to deduce

$$(\alpha\|\cdot\|_1 + \beta\|\cdot\|_2)^\circ = (\gamma_{\frac{1}{\alpha}\mathbb{B}_1} + \gamma_{\frac{1}{\beta}\mathbb{B}_2})^\circ = \gamma_{\left(\frac{1}{\alpha}\mathbb{B}_1\right)^\circ + \left(\frac{1}{\beta}\mathbb{B}_2\right)^\circ} = \gamma_{\alpha\mathbb{B}_\infty + \beta\mathbb{B}_2}.$$

A distinctive feature of all of the problems stated in Table 4.2 is the nondifferentiability of the objective of $\mathcal{Q}_\tau$. The choice seems especially peculiar when ρ is the 2-norm, since in that case, it is obvious that an *equivalent* smooth problem can be obtained by simply squaring the objective $\mathcal{Q}_\tau$ and the corresponding constraint in the original problem $\mathcal{P}_\sigma$. Of course, we do not prescribe the method for solving the level-set problem, and depending on the application and solvers available, it may be more convenient or efficient to solve a smooth variant of $\mathcal{Q}_\tau$ in order to obtain a solution of the nonsmooth version; cf. §4.3.1.

4.4.3 Generalized linear models

In all the examples we have seen so far, we have encountered only two types of misfit functions ρ, namely the squared 2-norm and the various gauges listed in Table 4.2. In this section, we broaden the scope by exploring several examples arising from statistical modeling. In particular, we consider the broad class of *generalized linear models* (GLMs) [64], which capture

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. $\varphi(x) \le \tau$	$-\varphi^\circ(A^T y)$
BPDN	$\min_x \|x\|_1$ s.t. $\|Ax-b\|_2 \le \sigma$	$\min_x \|Ax-b\|_2$ s.t. $\|x\|_1 \le \tau$	$-\|A^T y\|_\infty$
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. $\alpha\|x\|_1 + \beta\|x\|_2 \le \tau$	$-\gamma_{\alpha\mathbb{B}_\infty + \beta\mathbb{B}_2}(A^T y)$
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.

non-Gaussian data—including non-negative, count, boolean and multinomial variables—and robust log-concave densities.

GLMs assume that the observed data is distributed according to a member of the exponential family, and postulate a linear predictive model for key parameters. Suppose we are given data pairs $\{(b_i, a_i)\}_{i=1}^n \subset \mathbb{R} \times \mathbb{R}^m$, where b_i is an observation associated with the covariate vector a_i for individual $i = 1, \ldots, n$. GLMs assume that the postulated density for each response b_i is the function

$$p(b_i;\, \theta_i) = C(b_i, \phi) \exp\left(\frac{b_i \theta_i - c(\theta_i)}{\phi}\right), \tag{4.14}$$

where ϕ is the dispersion parameter, θ_i is the mean parameter, $c(\cdot)$ is a function that specifies the distribution, and $C(b_i, \phi)$ is a normalization constant that can depend on the data and ϕ. To simplify the exposition, we focus only on the canonical parameter θ_i, and assume that the dispersion parameter ϕ is known and present in its simplest form; see McCullah and Nelder [64] for more general cases. Whenever the function c is convex, it is clear that the resulting density is log-concave. To complete the GLM specification, one now assumes that b_i

is distributed according to the GLM in (4.14) with

$$\theta_i = a_i^T x,$$

where x is an unknown vector that is uniform across the population from which the data is selected. The task is to infer the vector x from the given data.

A technical concept in GLM modeling is the *link function*—an invertible function that maps likelihood parameters to the canonical parameter θ_i. For example, when working with count data, one encounters the Poisson distribution, which is proportional to $\exp(b_i \log \lambda_i - \lambda_i)$. We identify (4.14) with this distribution using the log link function, and set $\theta_i = \log \lambda_i$. It necessarily follows that $c(\theta_i) = \exp(\theta_i)$.

Assuming that the data is chosen independently from the population, the negative log-likelihood function for this model is given by

$$L(b; Ax) = \sum_{i=1}^{n} -\ln p(b_i; a_i^T x),$$

where a_i the ith row of the matrix A. The likelihood-constrained formulation $\mathcal{P}_\sigma$ for the regularized GLM is thus given by the problem

$$\underset{x}{\text{minimize}} \quad \varphi(x) \quad \text{subject to} \quad L(b; Ax) \leq \sigma, \tag{4.15}$$

where φ is a given regularizer. For example, the 1-norm regularizer may be used to induce sparsity in the parameter x. A reasonable choice for σ is a proportion of the expectation:

$$\sigma \propto \mathbb{E}\, L(b;\, Ax). \tag{4.16}$$

When an estimate of the expectation is not available, σ can be selected by using an expected variance-reduction scheme, so that $\sigma \propto L(b; 0)$, where the proportionality constant is chosen based on practitioner-prior experience.

Applying the level-set approach.

We now describe the various ingredients needed to apply the level-set approach to the GLM family. For simplicity, we assume that φ is a gauge, which captures a broad range of

Distribution	$c(\theta)$	link function	$c^\star(z)$
Gaussian	$\frac{1}{2}\theta^2$	id	$\frac{1}{2}z^2$
Huber [4]	$\rho_\kappa(\theta)$	id	$\frac{1}{2}z^2 + \delta_{\kappa\mathbb{B}_\infty}(z)$
Poisson	$\exp(\theta)$	log	$z\log z - z + \delta_{\mathbb{R}_+}(z)$
Bernoulli	$\log(1+\exp(\theta))$	logit	$z\log z + (1-z)\log(1-z) + \delta_{[0,1]}(z)$
Gamma	$-\log(-\theta)$	$(\cdot)^{-1}$	$1 - \log(-z) + \delta_{\mathbb{R}_-}(z)$

Table 4.3: Parameters of the GLM family, including required conjugates for their dual representation. The interpretation of coercive PLQ penalties (such as the Huber) as kernels of statistical distributions is developed in [4, Section 2].

regularizers (cf. §4.4.2). (Non-gauge regularizers are considered in §4.4.5.) The corresponding level-set problem $\mathcal{Q}_\tau$ is given by

$$\underset{x}{\text{minimize}} \quad L(b; Ax) \quad \text{subject to} \quad \varphi(x) \leq \tau. \tag{4.17}$$

In order to derive global affine minorants, we require the corresponding dual problem (cf. §4.2.3). Set $L_b(\cdot) := L(b;\cdot)$, and apply Fenchel duality to obtain

$$\underset{y}{\text{maximize}} \quad -L_b^\star(-y) - \tau\varphi^\circ(A^T y). \tag{4.18}$$

When p is as given in (4.14), we have $L_b(z) = K + \sum_i \phi^{-1}(c(z_i) - b_i z_i)$, where $K := -\sum_i \ln C(b_i;\phi)$. Hence, the dual problem takes the form

$$\underset{y}{\text{maximize}} \quad K - \frac{1}{\phi}\sum_i c^\star(b_i - \phi y_i) - \tau\varphi^\circ(A^T y). \tag{4.19}$$

Table 4.3 lists common exponential distributions and the link functions needed to represent them in the form of a GLM (4.14). The table also lists the resulting functions c and their conjugates needed for the dual.

A fair comparison of regularizers

Multiple experiments that involve different regularization functions can be easily compared at the same admissible levels of misfit using the formulation $\mathcal{P}_\sigma$. This feature of $\mathcal{P}_\sigma$ is unique among the alternative formulations.

As an example, consider classification using logistic regression (corresponding to the Bernoulli distribution) with either 1- or 2-norm regularization:

$$\underset{x}{\text{minimize}} \quad \|x\|_i \quad \text{subject to} \quad L(b; Ax) \leq \sigma, \tag{4.20}$$

for $i = 1, 2$. We set $\sigma := L(b; 0)/\eta$, where η is a specified proportionality constant. The likelihood of observing a Bernoulli random variable $b_i \in \{0, 1\}$ is given by

$$P(b_i) = \eta_i^{b_i}(1 - \eta_i)^{1-b_i},$$

where η_i is the probability of observing $b_i = 1$. Rewriting to match (4.14) gives

$$\begin{aligned} P(b_i) &= \exp\left(b_i \log(\eta_i) + (1 - b_i)\log(1 - \eta_i)\right) \\ &= \exp\left(b_i \log\left(\frac{\eta_i}{1 - \eta_i}\right) + \log(1 - \eta_i)\right), \end{aligned}$$

which identifies the link function from Table 4.3 with the canonical parameter $\theta_i = \log\left(\frac{\eta_i}{1-\eta_i}\right)$, and determines $c(\theta_i) = \log(1 + \exp(\theta_i))$. Composing with the linear model $\theta_i = a_i^T x$, we obtain the negative log likelihood objective (ignoring the constant term)

$$L(b; Ax) = \sum_{i=1}^{n} \log\left(1 + \exp(a_i^T x)\right) - b_i(a_i^T x).$$

We run the approach on the Adult dataset [59], which aims to predict whether people make more than \$50K a year. The challenge is that there are fewer positive than negative answers. The full dataset has $m = 122$ features and 48,844 individuals. We split this group into $n = 32562$ training and 16,282 test cases. In the test set, there are 3,846 individuals who make more than \$50K a year, and 12,436 who do not. Table 4.4 shows that the 1-norm regularization has as good or better generalizability at all tested levels of η. The 1-norm does as well or better than 2-norm with the cases (people earning more than \$50K), and gives a sparser model, while matching identification of controls (people earning less than \$50K).

μ	1.1	1.5	1.9	2.0	2.1
2-norm correct +	0	0.07	0.46	0.52	0.57
1-norm correct +	0	0.17	0.51	0.52	0.57
2-norm correct −	.96	0.98	0.94	0.94	0.93
1-norm correct −	.96	0.98	0.94	0.94	0.93
2-norm nonzero features	89	116	112	122	122
1-norm nonzero features	1	5	16	22	42

Table 4.4: Recovery results for likelihood-regularized $\mathcal{P}_\sigma$ logistic regression formulations. Fractions of correctly identified cases and controls in *test* set are shown for classifiers corresponding to optimal 2-norm and 1-norm solutions of (4.20), fitting the data in terms of the proportionality constant η.

Robust regression

As another example, we consider log-concave robust penalties—an important subclass of GLMs. We illustrate the modeling possibilities of this subclass, using the Huber penalty and its asymmetric extension, the quantile Huber (see Figure 4.3). The quantile Huber is parameterized by (κ, τ), which control the transition between quadratic and linear pieces, as well as the asymptotic slopes:

$$\rho_{\kappa,\tau}(r) \;=\; \begin{cases} \tau|r| - \frac{\kappa\tau^2}{2} & \text{if } r < -\tau\kappa, \\ \frac{1}{2\kappa}r^2 & \text{if } r \in [-\kappa\tau, (1-\tau)\kappa], \\ (1-\tau)|r| - \frac{\kappa(1-\tau)^2}{2} & \text{if } r > \ (1-\tau)\kappa. \end{cases} \tag{4.21}$$

The quantile Huber generalizes both the quantile loss and the Huber loss. We recover Huber when $\tau = 0.5$, and the quantile Huber converges to the quantile loss (known as the *check function*) as $\kappa \to 0$. When r is an m-vector instead of scalar, we write $\rho_{\kappa,\tau}(r) := \sum_{j=1}^m \rho_{\kappa,\tau}(r_j)$, and for simplicity we write $\rho_\kappa := \kappa\rho_{2\kappa,0.5}$ to denote the scaled Huber.

The Huber penalty figures prominently in high-dimensional regularized *robust* regression, as a measure of data misfit [21, 33, 36, 44, 57, 63]. High dimensional extensions (with sparse regularization) have been studied by Sun and Zhang [81] with applications to face recognition [91] and signal processing [46]. The quantile Huber, shown in Figure 4.3b, was recently introduced by Aravkin et al. [1] as an alternative to quantile regression—an asymmetric variant of the 1-norm used to analyze heterogeneous datasets [24, 49], such as those in computational biology [94], survival analysis [50], and economics [48, 51].

The methods of §8 allow one to easily explore robust regularization with the Huber penalty in the context of sparsity. Specifically, consider the BP_σ problem, but with the Huber penalty replacing the norm-squared error:

$$\underset{x}{\text{minimize}} \quad \|x\|_1 \quad \text{subject to} \quad \rho_{\kappa,\tau}(b - Ax) \leq \sigma. \tag{4.22}$$

It is well known that the Huber loss function is much less sensitive (i.e., robust) to outliers in the data than the norm-squared.

Example 4.4.1 (Robust sparse regression)**.** As a proof of concept, we illustrate the level-set framework on the following example. We generate a k-sparse signal of dimension $n \gg k$, measure it with $m = 5k$ Gaussian random vectors, and contaminate the measurements with asymmetric outliers. The results are shown in Figure 4.4. In the experiment, $n = 400$, $m = 100$, and $k = 10$. True measurements are obtained, and small Gaussian noise is added. The measurements are then contaminated by six positive outliers generated by sampling uniformly from $[0, 0.5]$. The 2-norm, symmetric Huber, and quantile Huber are compared using our proposed level-set framework; all models are fit to a level $\sigma = 0.05\rho(b)$, where b is the (contaminated) measurement vector. Both symmetric and quantile Huber show superior performance to the 2-norm. The advantage of the asymmetric Huber is fully evident in the residual plot. All the outliers in the example are positive, and using $\tau = 0.9$ for the quantile Huber, we identify all the outliers in the residual.

4.4.4 Low-rank matrix completion

A range of useful applications can be modeled as matrix completion problems. Important examples include applications in recommender systems and system identification (Recht, Fazel, Parillo [75]). The general principle 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. Applications include alignment of occluded images [74], scene triangulation [92], model selection [28], face recognition, and document indexing [25].

These problems can be formulated generally as

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimize}} \quad \varphi(X) \quad \text{subject to} \quad \rho(\mathcal{A}X - b) \leq \sigma, \tag{4.23}$$

where b is a vector of observations, the linear operator $\mathcal{A}$ encodes information about the measurement process, and the objective φ encourages the required structure in the solution, e.g., low-rank. The function ρ measures the misfit between the linear model $\mathcal{A}X$ and the observations b. If we wish to require $\mathcal{A}X = b$, we can simply set $\sigma = 0$ and choose any nonnegative convex function ρ with $\rho^{-1}(0) = \{0\}$, e.g., $\rho = ||\cdot||_2$. We categorize the problems of interest into two broad classes: *symmetric* and *asymmetric* problems. For each case, we outline how the level-set approach leads to implementable algorithms with computational kernels that scale gracefully with problem size.

The first class of problems aims to recover a low-rank PSD matrix, and in that case, the linear operator $\mathcal{A}$ maps between the space of symmetric $n \times n$ matrices and vectors, and we define the objective φ by

$$\varphi_1(X) = \operatorname{tr}(X) + \delta_{\mathcal{S}_+^n}(X).$$

Problem (4.23) then reduces to finding a minimum-trace, PSD matrix that satisfies the measurements specified by $\mathcal{A}X = b$. There are analogs for optimization over complex Hermitian matrices; we focus on the real case only for simplicity. The formulation above captures, for example, the *PhaseLift* approach to the phase-retrieval problem, which aims to

recover phase information about a signal (e.g., an image) by using only a series of magnitude measurements [27]. Important applications include optical wavefront reconstruction for astrophysical imaging [61] and the imaging of the molecular structure of a crystal via X-ray crystallography, which gives rise to such magnitude-only measurements; see Waldspurger, d'Aspremont, and Mallat [87] for a more complete description, including a number of other applications.

The second 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, and we define φ as the nuclear norm:

$$\varphi_2(X) = ||X||_* := \sum_{i=1}^{\min\{m,n\}} \sigma_i(X),$$

where $\sigma_i(X)$ is the ith singular value of X. This formulation captures, for example, the bi-convex compressed sensing problem [60].

Example 4.4.2 (Robust PCA)**.** The second class captures a range of problems that are not immediately of the form (4.23). For example, the stable version of the RPCA problem [90] aims to decompose a matrix Y as a sum of a low-rank matrix and a sparse matrix via the problem

$$\underset{L,S}{\text{minimize}} \quad \lambda\|L\|_* + \kappa\|S\|_1 + \tfrac{1}{2}\|\mathcal{A}(L-Y) - S\|_F^2. \tag{4.24}$$

Here the operator $\mathcal{A}$ is often a mask for the known elements of Y. The goal is to obtain a low-rank approximation to Y where the deviation from the known elements of Y is as sparse as possible. The parameters λ and κ are are chosen to balance the rank of L against the sparsity of the residual S while minimizing the least-squared misfit. This model can be given a statistical interpretation that fits nicely into the context of robust regression as presented in Section 4.4.3.

We proceed by eliminating S in (4.24) by first minimizing the objective over S alone—an overlooked algorithmic technique for this problem. Observe that, as a function of S, the objective is the Moreau envelope of the 1-norm evaluated at $\mathcal{A}(L-Y)$, or, equivalently, the

Huber function ρ_κ on $\mathbb{R}^{m\times n}$ (4.21):

$$\inf_S \left\{\kappa\|S\|_1 + \tfrac{1}{2}\|\mathcal{A}(L-Y)-S\|_F^2\right\} = \rho_\kappa(\mathcal{A}(L-Y)).$$

Problem (4.24) can now be written in terms of L alone:

$$\underset{L}{\text{minimize}} \quad \lambda\|L\|_* + \rho_\kappa(\mathcal{A}(L-Y)).$$

This is the Lagrangian form of the robust estimation problem (4.22). Arguably, we can now interpret the goal of this problem as one of finding the lowest rank approximation to Y over its known elements subject to a bound on a robust measure of misfit. This yields the problem

$$\underset{L}{\text{minimize}} \quad ||L||_* \quad \text{subject to} \quad \rho_\kappa(\mathcal{A}(L-Y)) \leq \sigma, \tag{4.25}$$

for some choice of parameter $\sigma \geq 0$. Various principled choices for σ are discussed in §4.4.3.

Level-set approach and the Frank-Wolfe oracle

We apply the level-set approach, and exchange the roles of the regularizing function φ and the misfit $\rho(\mathcal{A}X - b)$. Note that 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. Therefore, for both cases, we may use the first row of Table 4.2 to determine the corresponding level-set subproblem and affine minorants based on dual certificates. In particular, the corresponding level-set subproblem $\mathcal{Q}_\tau$, which defines the value function, is

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

We use the polar calculus described by Friedlander et al. [40, §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 and asymmetric 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. As numerous authors have observed, the Frank-Wolfe algorithm [39, 45] is therefore especially well suited for evaluating the required quantities, and here we describe how to apply the algorithm to this setting.

The Frank-Wolfe subproblem (4.5), used to generate search directions at each iteration, takes the form

$$\underset{S}{\text{maximize}} \ \langle G, S \rangle \quad \text{subject to} \quad \varphi(S) \leq \tau. \tag{4.26}$$

where $G := \mathcal{A}^* \nabla \rho(\mathcal{A}X - b)$ is the gradient of $\rho(\mathcal{A}X - b)$ 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 $[\varphi \leq \tau]$ and the ray $X + \mathbb{R}_+(S - X)$.

Solutions for the linearized subproblems can be obtained by computing extreme eigenvalues or singular values of G [45, §4.2]. For the symmetric case, the constraint

$$\varphi_1(S) \leq \tau \qquad \text{is equivalent to} \qquad \operatorname{tr}(S) \leq \tau, \ S \succeq 0.$$

The linearized subproblem (4.26) is then solved by any matrix of the form

$$S = U \operatorname{Diag}(\xi_i) U^T \quad \text{with} \quad \sum_{i=1}^{k} \xi_i = \tau, \ \xi_i \geq 0,$$

where $U \in \mathbb{R}^{n \times k}$ is the matrix that collects the k eigenvectors of G corresponding to $\lambda_1(G)$. For the non-symmetric case, the constraint $\varphi_2(S) \leq \tau$ is simply $||S||_* \leq \tau$, and the linearized subproblem is solved by any matrix of the form

$$S = U \operatorname{Diag}(\xi_i) V^T \qquad \text{with} \qquad \sum_{i=1}^{k} \xi_i = \tau, \ \xi_i \geq 0,$$

where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ are the matrices that collect the k singular vectors of G corresponding to the leading singular value $\sigma_1(G)$. In both cases, Krylov-based eigensolvers, such as ARPACK [53] 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 (FFT). The following example illustrates exactly this scenario.

Example 4.4.3 (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

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

In literature, this problem is also often called ℓ_2 graph embedding and appears in wireless networks, statics, robotics, protein reconstruction, and manifold learning; see the recent survey [58].

A popular convex relaxation for this problem was introduced by Weinberger et al. [88], and extensively studied by a number of authors [14, 16, 34]:

$$\begin{array}{ll} \text{maximize} & \operatorname{tr} X \\ \text{subject to} & \|\mathcal{P}_E \circ \mathcal{K}(X) - \omega\| \leq \sigma, \\ & Xe = 0,\ X \succeq 0, \end{array} \tag{4.27}$$

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 $\mathcal{P}_E(D)$ is the canonical projection of a matrix D 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. It is then easy to see that the rows of P are the points $p_1, \ldots, p_n \in \mathbb{R}^r$ we seek. The constraint $Xe = 0$ simply ensures that the points p_i are centered around the origin. Notice, that this formulation directly contrasts the usual min-trace regularizer in compressed sensing; nonetheless, it is

very natural. An easy computation shows that in terms of any factorization $X = PP^T$, the equality $\operatorname{tr}(X) = \frac{1}{2n}\sum_{i,j=1}^{n} \|p_i - p_j\|^2$ holds. Thus trace maximization serves to "flatten" the realization of the graph.

It is known that for $\sigma = 0$, the problem formulation (4.27) notoriously fails strict feasibility [34, 35, 52]. In particular, for small $\sigma \geq 0$ the feasible region is very thin and the solution to the problem is unstable. As a result, algorithms maintaining feasibility are likely to exhibit some difficulties. In contrast, following the theme of this paper, we employ 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 (4.27) is

$$\begin{array}{ll} \text{minimize} & \|\mathcal{P}_E \circ \mathcal{K}(X) - \omega\| \\ \text{subject to} & \operatorname{tr} X \geq \tau,\ Xe = 0,\ X \succeq 0. \end{array} \tag{4.28}$$

Note the direction of the inequality $\operatorname{tr} X \geq \tau$, which takes into account that the original formulation (4.27) is a *maximization* problem. As a result, the root-finding method on the value function will approach the optimal value $\tau_* = \mathrm{OPT}$ from the right. In particular, to initialize the approximate Newton scheme, we need an upper bound τ_0 on the objective function. Such upper bounds are easily available from the diameter of the graph. See Figure 4.5 for an illustration.

Note that the gradient of the objective function is typically very sparse (as sparse as the edge set E). Moreover, the linear subproblem over the feasible region is analogous to the ones considered in Section 4.4.4, requiring only a maximal eigenvalue computation on a sparse matrix (the gradient of the objective function); for more details see [34]. This makes the problem (4.28) ideally suited for the Frank-Wolfe algorithm, as discussed in §4.4.4. We note that the dual problem of (4.28) takes the form

$$\underset{y \in \mathbb{R}^E,\ \|y\|_2 \leq 1}{\text{maximize}} \quad \langle y, \omega \rangle - 2\tau \lambda_{\max}^{e^\perp}(\operatorname{Diag}(Ye) - Y).$$

The matrix $Y = \mathcal{P}_E^*(y)$ is the vector y padded with zeros and then $2(\operatorname{Diag}(Ye) - Y) = \mathcal{K}^*\mathcal{P}_E^*(y)$. The symbol $\lambda_{\max}^{e^\perp}(A)$ is the maximal eigenvalue of the restriction of the matrix A to $e^\perp$. Hence,

$\mathcal{P}_\sigma$	$\mathcal{Q}_\tau$	Dual of $\mathcal{Q}_\tau$
$\min_x \varphi_{en}(x)$	$\min_x \rho_\kappa(Ax-b)$	$\max_{y\in\kappa\mathbb{B}_\infty} \langle b,\, y\rangle - \frac{\kappa}{2}\|y\|_2^2 - \delta^\star_{[\varphi_{en}\le\tau]}(A^T y)$
s.t. $\rho_\kappa(Ax-b)\le\sigma$	s.t. $\varphi_{en}(x)\le\tau$	

Table 4.5: Elastic net

affine minorants are immediate to read off from the dual certificates generated by the Frank-Wolfe algorithm. An extensive numerical investigation of this approach is made by Drusvyatskiy et al. [34].

4.4.5 Robust elastic net regularization

In this final section, we explore an important data fitting problem where the regularizer φ is not a gauge, unlike our previous examples. Zou and Hastie [93] introduced the *elastic net regularizer*

$$\varphi_{en}(x) := \alpha\|x\|_1 + \frac{1-\alpha}{2}\|x\|_2^2 \qquad (0\le\alpha\le 1)$$

for situations where there are multiple groups of covariates that are strongly correlated within each group. In this setting, the LASSO typically picks one member from each of the most important groups whereas the elastic net can pick out both the important groups and their members. As is the case with the Huber function, φ_{en} is a member of the PLQ family [3, 4].

Zou and Hastie only consider the LS_τ and QP_λ formulations of the 1-norm regularized problem discussed in §4.1, but with $\|\cdot\|_1$ replaced by φ_{en}. Furthermore, they focus on the Lagrangian formulation QP_λ for computational reasons. The problem corresponding to BP_σ is not investigated. In this section, we provide a guide to the implementation of the methods of §4.2 for this version of the elastic net problem, but generalized to the case where the residual term is replaced by the Huber function ρ_κ in (4.21) for robust inference. This gives the three formulations described in Table 4.5, which we call the *robust elastic net problem*.

Inexact oracle for the value function

From Table 4.5, we determine the value function

$$v(\tau) := \min \{\rho_\kappa(Ax - b) \mid \varphi_{en}(x) \leq \tau\}$$

to which we apply the root-finding procedure. We solve $\mathcal{Q}_\tau$ via an optimal gradient-projection algorithm, as described in §4.4.1. The methods require at each iteration a projection onto the level sets $[\varphi_{en} \leq \tau]$, which is given as the solution of the problem

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|x - z\|_2^2 \quad \text{subject to} \quad \varphi_{en}(x) \leq \tau. \tag{4.29}$$

The projection problem can be solved as follows. Assume without loss of generality $z \notin [\varphi_{en} \leq \tau]$ since otherwise $x := z$ solves (4.29). We may also assume—possibly after a coordinate sign change—that $z \geq 0$. Observe then that any optimal solution x satisfies $x \geq 0$. Thus, a feasible point x solves (4.29) if and only if there exists a scalar $\lambda > 0$ that satisfies

$$0 \in (x - z) + \lambda(1 - \alpha)x + \lambda\alpha\partial\|\cdot\|_1(x).$$

Equivalently,

$$z \in (1 + \lambda(1 - \alpha))x + \lambda\alpha\partial\|\cdot\|_1(x),$$

which amounts to the coordinate-wise inclusion

$$z_i \in (1 + \lambda(1 - \alpha))x_i + \lambda\alpha\partial|\cdot|(x_i) \quad \text{for each } i = 1, \ldots, n.$$

In the case $x_i = 0$, simple arithmetic shows $(z_i - \lambda\alpha\mathsf{sgn}(z_i))_+ = 0$. Otherwise when $x_i \neq 0$, the numbers x_i and z_i are both strictly positive, and

$$x_i = \frac{(z_i - \lambda\alpha\mathsf{sgn}(z_i))_+}{1 + \lambda(1 - \alpha)} = \frac{(z_i - \lambda\alpha)_+}{1 + \lambda(1 - \alpha)}. \tag{4.30}$$

Hence, regardless of whether x_i is zero or not, (4.30) holds for all $i = 1, \ldots, n$. Plugging this into the relation $\varphi_{en}(x) = \tau$ gives

$$\begin{aligned} \tau &= \alpha \sum_i \frac{(z_i - \lambda\alpha)_+}{1 + \lambda(1 - \alpha)} + \frac{(1 - \alpha)}{2} \sum_i \frac{(z_i - \lambda\alpha)_+^2}{(1 + \lambda(1 - \alpha))^2} \\ &= \frac{\alpha}{1 + \lambda(1 - \alpha)} \sum_i (z_i - \lambda\alpha)_+ + \frac{(1 - \alpha)}{2(1 + \lambda(1 - \alpha))^2} \sum_i (z_i - \lambda\alpha)_+^2. \end{aligned} \tag{4.31}$$

The strong convexity of the objective in x implies that there is a unique positive λ that solves this equation. In addition, for $\lambda \geq \alpha^{-1} \|z\|_\infty$, the right-hand side of (4.31) is zero, while for $\lambda = 0$, the right-hand side is $\varphi(z) > \tau$. So the unique optimal λ resides in the open interval $(0, \alpha^{-1} \|z\|_\infty)$. Finally, since $(1 + \lambda(1-\alpha)) > 0$ for all $\lambda \geq 0$, equation (4.31) is equivalent to

$$0 = \tau(1 + \lambda(1-\alpha))^2 - \alpha(1 + \lambda(1-\alpha)) \sum_i (z_i - \lambda\alpha)_+ - \frac{(1-\alpha)}{2} \sum_i (z_i - \lambda\alpha)_+^2.$$

The root λ is found by sorting coordinates of z and then solving a quadratic polynomial in λ. Substituting λ back into (4.30), we find the optimal x.

Affine minorant oracle for the value function

Following the approach of §4.2.3, for each candidate value of τ in Algorithm 8, we generate a dual certificate y that yields a lower-bound on the value function $v(\tau)$. Such dual iterates are generated automatically by fast gradient methods on the primal problem [82]. To obtain an affine minorant of v, we then need a method for evaluating the function

$$\Phi(y, \tau) := \langle b,\, y \rangle - \tfrac{1}{2} \|y\|_2^2 - \delta^\star_{[\varphi_{en} \leq \tau]}(A^T y),$$

and a subgradient $s \in \partial_\tau \delta^\star_{[\varphi_{en} \leq \tau]}(A^T y)$. To this end, we use the representation

$$\delta^\star_{[\varphi_{en} \leq \tau]}(z) = \inf_{\mu > 0} [\tau\mu + \mu \varphi^\star_{en}(\mu^{-1} z)].$$

See, for example, Aravkin et al. [3, Equation 6.5c]. Since φ_{en} is the sum of two finite-valued convex functions, its conjugate is the infimal convolution

$$\varphi^\star_{en}(z) = \inf_{v \in \alpha\mathbb{B}_\infty} \frac{1}{2(1-\alpha)} \|z - v\|_2^2 = \frac{1}{2(1-\alpha)} \mathrm{dist}^2_{\alpha\mathbb{B}_\infty}(z) = \frac{1}{2(1-\alpha)} \|(|z| - \alpha e)_+\|_2^2 .$$

Hence, for $\mu > 0$, we have

$$\delta^\star_{[\varphi_{en} \leq \tau]}(z) = \inf_{\mu > 0} \left\{ \tau\mu + \frac{1}{2(1-\alpha)\mu} \|(|z| - \mu\alpha \mathrm{e})_+\|_2^2 \right\}, \tag{4.32}$$

and the derivative of $\delta^\star_{[\varphi_{en} \leq \tau]}(z)$ with respect to τ is given by the optimal μ when it exists. Note that if $\mu \geq \alpha^{-1} \|z\|_\infty$, then $[\tau\mu + \frac{1}{2(1-\alpha)\mu} \|(|z| - \mu\alpha \mathrm{e})_+\|_2^2] = \tau\mu$, while for $\mu \to 0$ we

have $[\tau\mu + \frac{1}{2(1-\alpha)\mu}\|(|z| - \alpha\mu e)_+\|_2^2] \to +\infty$. Hence, an optimal μ exists when $\tau > 0$. It is also unique due to the convex piecewise quadratic nature of the objective. Consequently, the optimal μ in (4.32) can be obtained by sorting $|z|$ and then writing in closed form the solution of a sequence of elementary univariate convex functions over an interval.

4.5 Proofs

Theorem 4.5.1 (Superlinear convergence of Newton and secant methods)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a non-increasing, convex function on the interval $[a, b]$. Suppose that the point $\tau_* := \inf\{\tau : f(\tau) \le 0\}$ lies in (a, b) and the non-degeneracy condition $g_* := \inf\{g \mid g \in \partial f(\tau_*)\} < 0$ holds. Fix two points $\tau_{-1}, \tau_1 \in (a, b)$ satisfying $\tau_0 < \tau_1 < \tau_*$ and consider the following two iterations:*

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

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 $|\tau_* - \tau_{k+1}| \le (1 - \frac{g_*}{\gamma_k})|\tau_* - \tau_k|$, $k = 1, 2, \ldots$, 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.

Proof. 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

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

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 $g_0 \leq g_1 \leq g_2 \leq \cdots \leq g_* < 0$. We have

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

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

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

Combining the two inequalities yields

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

Consequently, we deduce

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

The result follows. □

Proof of Theorem 4.2.1. It is easy to see by convexity that the iterates τ_k are strictly increasing and satisfy $f(\tau_k) > 0$. For each index $j \geq 2$, define the following quantities:

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

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, $0 \leq \theta_j \leq \alpha - \gamma_j$, is valid. Define now constants $\beta_j \in [0, 1]$ by the equation $\gamma_j = \beta_j \alpha$. Suppose $k \geq 2$ is an index at which the algorithm has not terminated, i.e., $u_k > \epsilon$. Taking into account the inequality $\ell_k \geq \frac{u_k}{\alpha} > \frac{\epsilon}{\alpha}$, we deduce

$$\frac{\epsilon}{\alpha} \leq \ell_k = \ell_1 \prod_{j=2}^{k} \gamma_j \leq C\alpha^{k-1} \prod_{j=2}^{k} \beta_j. \tag{4.33}$$

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 $\tau_* - \tau_1 \geq h_{k+1}$, $\ell_k \geq \frac{\epsilon}{\alpha}$, and $\theta_j \leq \alpha - \gamma_j$ imply

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

and rearranging gives

$$\epsilon \leq (\tau_* - \tau_1)|s_1|\alpha^k \prod_{j=2}^{k}(1-\beta_j) \leq C\alpha^k \prod_{j=2}^{k}(1-\beta_j). \tag{4.34}$$

Combining (4.33) and (4.34), we get

$$\epsilon \leq C\alpha^k \min\left\{\prod_{j=2}^{k}\beta_j,\ \prod_{j=2}^{k}(1-\beta_j)\right\}. \tag{4.35}$$

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

$$\min\left\{\prod_{j=2}^{k}\beta_j,\ \prod_{j=2}^{k}(1-\beta_j)\right\} \leq 0.5^{k-1}. \tag{4.36}$$

Combining equations (4.36) and (4.35), the claimed estimate $k-1 \leq \log_{2/\alpha}\left(\frac{\alpha C}{\epsilon}\right)$ follows. □

Proof of Theorem 4.2.2. The proof is identical to the proof of Theorem 4.2.1, 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 v, 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 4.5.2 (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. Convexity of f immediately yields the inclusion

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

We deduce $\lambda \cdot \delta^*_{[f \leq a]}(z) + (1-\lambda) \cdot \delta^*_{[f \leq b]}(z) = \delta^*_{\lambda \cdot [f \leq a] + (1-\lambda) \cdot [f \leq b]}(z) \leq \delta^*_{[f \leq \lambda a + (1-\lambda)b]}(z)$, and the result follows. □

Proof of Proposition 4.3.1. 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 (4.4) 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

$$\frac{u-\sigma}{\hat{\ell}-\sigma} = \frac{(u-\sigma)\|y\|}{\Phi_2(y,\tau) + \frac{1}{2}\|y\|^2 - \sigma\|y\|} = \frac{2(u-\sigma)\|y\|}{\ell^2 + \|y\|^2 - 2\sigma\|y\|}. \tag{4.37}$$

Taking into account the equivalence

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

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 (4.37) 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 (4.37). □

Lemma 4.5.3. $(-\lambda_{\min})^\star(y) = \delta_{\mathcal{S}}(-y)$, *where* $\mathcal{S} = \mathcal{K}^* \cap \{x \mid \langle e,\, x\rangle = 1\}$.

Proof. The following formula is established in [77]:

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

or equivalently

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

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 $\operatorname{range}\partial(-\lambda_{\min}) = -\mathcal{S}$. 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. □

Lemma 4.5.4. *Let* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *be two nonempty closed convex sets that contain the origin. Then* $\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2} = \gamma_{(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ)^\circ}$. *If additionally* $0 \in \operatorname{int}\mathcal{D}_1$, *then* $(\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2})^\circ = \gamma_{\mathcal{D}_1^\circ + \mathcal{D}_2^\circ}$.

Proof. Theorem 14.5 of [78] contains most of the needed tools. In particular, the gauge of any closed convex function containing the origin is the support function of the polar. Thus,

$$\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2} = \delta^*_{\mathcal{D}_1^\circ} + \delta^*_{\mathcal{D}_2^\circ}.$$

By [43, Cor. 3.2.5], we have

$$\delta^*_{\mathcal{D}_1^\circ} + \delta^*_{\mathcal{D}_2^\circ} = \delta^*_{\mathrm{cl}\left(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ\right)} = \delta^*_{\mathcal{D}_1^\circ + \mathcal{D}_2^\circ},$$

where the last equality holds because the support function does not distinguish a set from its closure. Again using the polarity correspondence between the gauge and support functions, we have $\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2} = \gamma_{(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ)^\circ}$, as required. We now prove the second part of the lemma. Use the first part of the result and [78, Thm. 15.1] to deduce that

$$(\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2})^\circ = \gamma_{(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ)^{\circ\circ}}. \tag{4.38}$$

Because $0 \in \operatorname{int} \mathcal{D}_1$, the set $\mathcal{D}_1^\circ$ is compact [78, Cor. 14.5], and because $\mathcal{D}_2^\circ$ is closed, $\mathcal{D}_1^\circ + \mathcal{D}_2^\circ$ is also closed. Thus, $(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ)^{\circ\circ} = \mathcal{D}_1^\circ + \mathcal{D}_2^\circ$. It then follows from (4.38) that $(\gamma_{\mathcal{D}_1} + \gamma_{\mathcal{D}_2})^\circ = \gamma_{(\mathcal{D}_1^\circ + \mathcal{D}_2^\circ)}$, as required. □

Remark 3 (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

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

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

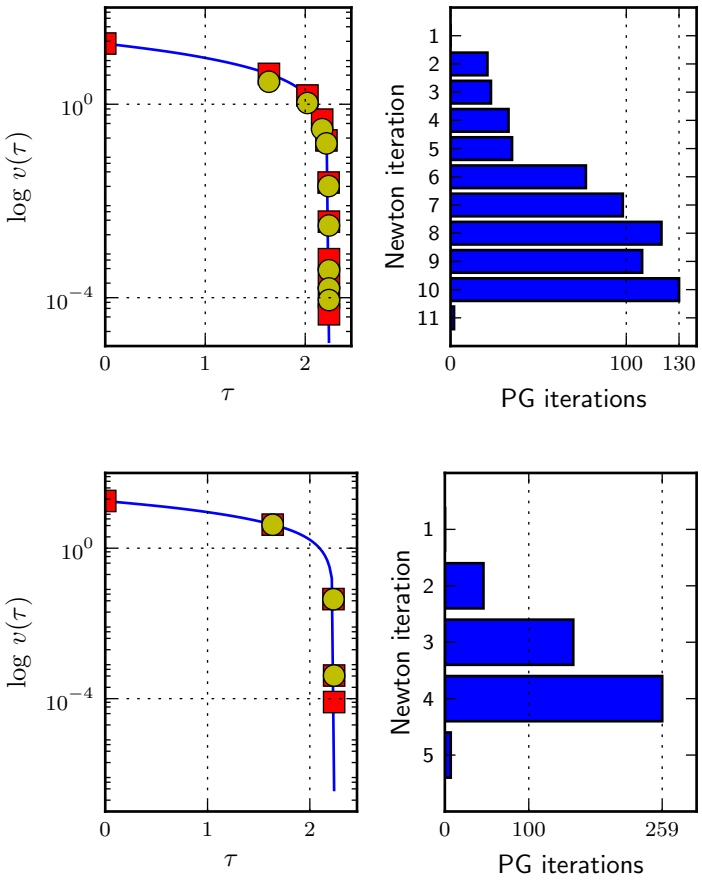
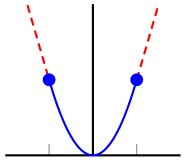
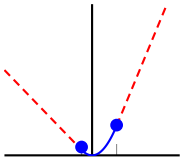


Figure 4.2: Progress of the root-finding method for a linear program. The panels on the left depict the graph of $v(\tau)$ (solid line), and the squares and circles, respectively, show the upper and lower bounds computed using an optimal projected-gradient method. The horizontal log scale results in 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: $\alpha = 1.01$.

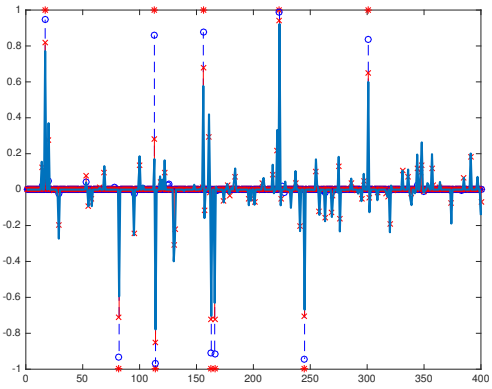


(a) Huber, $\kappa = 1$

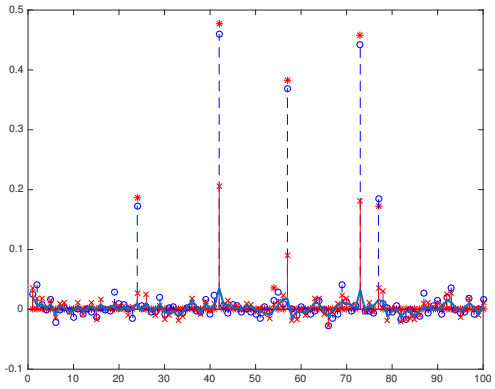


(b) quantile Huber

Figure 4.3: Huber penalty and its asymmetric extension, quantile Huber.



(a) True and Fitted signals. Red asterisks show true sparse signal; blue solid line shows LS estimate; solid dashed line ending in 'x' shows Huber estimate; thin dashed line ending with 'o' marker shows quantile Huber estimate.



(b) True and Fitted Residuals. Red asterisk shows true outliers; blue solid line shows LS residual; solid dashed line ending in 'x' shows Huber residual; thin dashed line ending with 'o' marker shows quantile Huber residual.

Figure 4.4: Robust Asymmetric Recovery, comparing 2-norm, Huber, and quantile Huber penalties. $\kappa = 0.1$ for both Huber penalties; quantile Huber has $\tau = 0.9$ to capture the fact that outliers are expected to be positive.

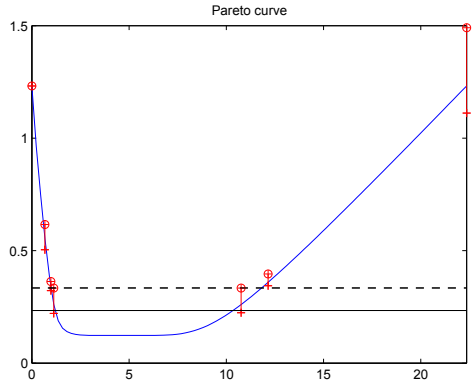


Figure 4.5: Value function $v(\tau) := \inf \{\|\mathcal{P}_E \circ \mathcal{K}(X) - \omega\| \mid \operatorname{tr} X = \tau,\ Xe = 0,\ X \succeq 0\}$. Newton's method converges to either the minimum- or maximum-trace solution, depending on if it is started with an iterate to the left of the minimal root, or to the right of the maximal root. For a solution of (4.27), we require the maximal root. In this experiment, $\sigma = 0.25$.

BIBLIOGRAPHY

[1] A. Aravkin, P. Kambadur, A.C. Lozano, and R. Luss. Orthogonal matching pursuit for sparse quantile regression. In *Data Mining (ICDM), International Conference on*, pages 11–19. IEEE, 2014.

[2] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *Preprint, arXiv:1407.1537*, 2016.

[3] A. Y. Aravkin, J. Burke, and M. P. Friedlander. Variational properties of value functions. *SIAM J. Optimization*, 23(3):1689–1717, 2013.

[4] Aleksandr Aravkin, James V. Burke, and Gianluigi Pillonetto. Sparse/robust estimation and kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory. *Journal of Machine Learning Research*, 14:2689–2728, 2013.

[5] H. Attouch, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Math. Program., doi:10.1007/s10107-016-0992-8*, pages 1–53, 2016.

[6] F. Bach. Duality between subgradient and conditional gradient methods. *SIAM J. Optim.*, 25(1):115–129, 2015.

[7] A. Beck. On the convexity of a class of quadratic mappings and its application to the problem of finding the smallest ball enclosing a given intersection of balls. *J. Global Optim.*, 39(1):113–126, 2007.

[8] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[9] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[10] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2001. Analysis, algorithms, and engineering applications.

[11] Dimitri P. Bertsekas. *Convex optimization algorithms*. Athena Scientific, Massachusetts, 2015.

[12] José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pages 1–4. IEEE, 2010.

[13] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

[14] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *Automation Science and Engineering, IEEE Transactions on*, 3(4):360–371, Oct 2006.

[15] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

[16] P. Biswas and Y. Ye. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale optimization methods and applications*, volume 82 of *Nonconvex Optim. Appl.*, pages 69–84. Springer, New York, 2006.

[17] Jonathan M. Borwein and Adrian S. Lewis. *Convex analysis and nonlinear optimization*, volume 3 of *CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC*. Springer, New York, second edition, 2006. Theory and examples.

[18] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[20] Peter Brucker. An o(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163 – 166, 1984.

[21] K.P. Bube and T. Nemeth. Fast line searches for the robust solution of linear systems in the hybrid ℓ_1/ℓ_2 and huber norms. *Geophysics*, 72(2):A13–A17, 2007.

[22] S. Bubeck and Y.T. Lee. Black-box optimization with a politician. *Preprint, arXiv:1602.04847*, 2016.

[23] S. Bubeck, Y.T. Lee, and M. Singh. A geometric alternative to Nesterov's accelerated gradient descent. *Preprint, arXiv:1506.08187*, 2015.

[24] Moshe Buchinsky. Changes in the u.s. wage structure 1963-1987: Application of quantile regression. *Econometrica*, 62(2):405–58, March 1994.

[25] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. Assoc. Comput. Mach.*, 58(3):1–37, May 2011.

[26] E.J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Info. Th.*, 56(5):2053–2080, 2010.

[27] Emmanuel J Candès, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Commun. Pur. Appl. Ana.*, 2012.

[28] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *Ann. Stat.*, 40(4):1935–2357, 2012.

[29] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[30] Peter Robert Chares. *Cones and Interior-Point Algorithms for Structured Convex Optimization involving Powers and Exponentials*. PhD thesis, Universite catholique de Louvain, 2007.

[31] S. Chen and S. Ma. Geometric descent method for convex composite minimization. *arXiv:1612.09034*, 2017.

[32] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1999.

[33] D. I. Clark. The mathematical structure of huber's m-estimator. *SIAM journal on scientific and statistical computing*, 6(1):209–219, 1985.

[34] D. Drusvyatskiy, N. Krislock, Y.-L. Voronin, and H. Wolkowicz. Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. *Preprint, arXiv:1410.6852*, 2014.

[35] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.*, 25(2):1160–1178, 2015.

[36] Rudolf Dutter and Peter J. Huber. Numerical methods for the nonlinear robust regression problem. *Journal of Statistical Computation and Simulation*, 13:79–113, 1981.

[37] R. h. Ennis and G. C. McGuire. *Computer Algebra Recipes: A Gourmet's Guide to the Mathematical Models of Science*. Springer, 2001.

[38] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Elec. Eng. Dept, Stanford University, 2002.

[39] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 3:95–110, 1956.

[40] M.P. Friedlander, I. Macêdo, and T.K. Pong. Gauge optimization and duality. *SIAM J. Optim.*, 24(4):1999–2022, 2014.

[41] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications*, 2(1):17–40, 1976.

[42] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Math. Program.*, 152(1-2, Ser. A):75–112, 2015.

[43] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, New York, NY, USA, 2001.

[44] P. J. Huber. *Robust Statistics*. John Wiley and Sons, 2 edition, 2004.

[45] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proc. 30th Intern. Conf. Machine Learning (ICML-13)*, pages 427–435, 2013.

[46] Vassilis Kekatos and Georgios B Giannakis. From sparse signals to sparse residuals for robust sensing. *Signal Processing, IEEE Transactions on*, 59(7):3355–3368, 2011.

[47] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[48] R. Koenker. *Quantile Regression*. Cambridge University Press, 2005.

[49] R. Koenker and G Bassett. Regression quantiles. *Econometrica*, pages 33–50, 1978.

[50] R. Koenker and O. Geling. Reappraising medfly longevity: A quantile regression survival analysis. *Journal of the American Statistical Association*, 96:458–468, 2001.

[51] Roger Koenker and Kevin F. Hallock. Quantile regression. *Journal of Economic Perspectives, American Economic Association*, pages 143–156, 2001.

[52] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM J. Optim.*, 20(5):2679–2708, 2010.

[53] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK Users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. SIAM, 1998.

[54] C. Lemaréchal. An extension of Davidon methods to nondifferentiable problems. *Math. Programming Stud.*, 3:95–109, 1975.

[55] C Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Programming*, 69(1, Ser. B):111–147, 1995. Nondifferentiable and large-scale optimization (Geneva, 1992).

[56] L. Lessard, B. Recht, and A. Packard. Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints. *SIAM J. Optim.*, 26(1):57–95, 2016.

[57] W. Li and J. Swetits. The linear l1 estimator and the huber m-estimator. *SIAM Journal on Optimization*, 8(2):457–475, 1998.

[58] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[59] M. Lichman. UCI machine learning repository, 2013.

[60] Shuyang Ling and Thomas Strohmer. Self-calibration and biconvex compressive sensing. *CoRR*, abs/1501.06864, 2015.

[61] Russell Luke, James Burke, and Richard Lyons. Optical wavefront reconstruction: theory and numerical methods. *SIAM Review*, 44:169–224, 2002.

[62] H. M. Markowitz. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Frank J. Fabozzi Associates, New Hope, Pennsylvania, 1987.

[63] R.A. Maronna, D. Martin, and V.J. Yohai. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, 2006.

[64] P McCullagh and J A Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1989.

[65] K. Miettinen. *Nonlinear Multi-Objective Optimization*. Springer Science+Business Media, New York, 1999.

[66] A. Nemirovski. 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 (electronic), 2004.

[67] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numer.*, 17:191–234, 2008.

[68] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[69] Yu. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[70] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[71] Yurii Nesterov. Towards non-symmetric conic optimization. *Optim. Methods Softw.*, 27(4-5):893–917, 2012.

[72] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[73] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[74] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2233–2246, 2012.

[75] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

[76] J. Renegar. Linear programming, complexity theory and elementary functional analysis. *Math. Programming*, 70(3, Ser. A):279–351, 1995.

[77] J. Renegar. A framework for applying subgradient methods to conic optimization problems. *Preprint arXiv:1503.02611 [math.CA]*, 2015.

[78] R T Rockafellar. *Convex Analysis*. Priceton Landmarks in Mathematics. Princeton University Press, 1970.

[79] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1998.

[80] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterovs accelerated gradient method: Theory and insights. In Z. Ghahramani, M. Welling, C. Cortes, N.d.

Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2510–2518. Curran Associates, Inc., 2014.

[81] Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, page ass043, 2012.

[82] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, University of Washington, 2008.

[83] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Program.*, 125:263–295, 2010.

[84] Levent Tunçel and Arkadi Nemirovski. Self-concordant barriers for convex approximations of structured convex sets. *Found. Comput. Math.*, 10(5):485–525, 2010.

[85] E. van den Berg and M. P. Friedlander. Sparse optimization with least-squares constraints. *SIAM J. Optimization*, 21(4):1201–1229, 2011.

[86] E. van den Berg and M.P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[87] Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Math. Prog.*, 149(1-2):47–81, 2015.

[88] K.Q. Weinberger, F. Sha, and Lawrence K. S. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 106–, New York, NY, USA, 2004. ACM.

[89] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Math. Programming Stud.*, 3:145–173, 1975.

[90] J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *Neural Information Processing Systems (NIPS)*, 2009.

[91] Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Robust sparse coding for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 625–632. IEEE, 2011.

[92] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: Transform invariant low-rank textures. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *Computer Vision – ACCV 2010*, volume 6494 of *Lecture Notes in Computer Science*, pages 314–328. Springer, 2011.

[93] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[94] Hui Zou and Ming Yuan. Regularized simultaneous model selection in multiple quantiles regression. *Computational Statistics & Data Analysis*, 52(12):5296–5304, 2008.