Optimization with Costly Subgradients

By

Gabriel Goh

B.S. (Simon Fraser University) 2001

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Michael Friedlander

Jesus De Loera

Matthias Koeppe

Committee in Charge

2017

To my Family, and Marcos.

Contents

Gabriel Goh
May 2017

Abstract

This thesis concerns the exploitation of two related classes of convex optimization problems that arise frequently in practice. The first are problems of two distinct blocks of variables, $h(x, y)$ where the objective is written implicitly as either the minimization or maximization over the second block. Problems with this structure present immediate computational challenges: any computation of a subgradient, or for that matter the objective, requires the solution of a convex program. While this may make the problem appear intractable superficially, it is often the case that a precise computation of first-order information is an overkill. Indeed, a suboptimal solution will yield an approximate subgradient, and when used in conjunction with a cutting-plane algorithm, yields a shallow-cut. When tuned appropriately, linear convergence, which is a property of the idealized algorithm, can still be preserved: this is the shallow-cut cutting-plane algorithm. In this thesis, we propose an alternative to this classical algorithm : the epigraph inexact cutting-plane method. This method allows us to use two approximate cuts at every step, one from an upper bound on the optimal solution, and one from a lower bound on f, an approximate subgradient, both found via duality. This technique makes full use of all the information obtained from an approximate solution and it is also far easier to quantify the total amount of work needed for convergence in terms of both the number of cuts used and the amount of effort spent in computing the first-order approximation.

The second class of structures of interest are those with an additive structure, $f + g$, where f is smooth and g is convex and simple. Furthermore, we assume f can be written as an expectation over some random variable. In a typical application, f is a loss term, taken over data, and g is a regularizer. The standard tool for solving problems with composite structure is the proximal-gradient algorithm. Its convergence, unfortunately, depends strongly upon the conditioning of the problem, and stalls in many real-world situations. This issue can be mitigated in part by using curvature approximation. Quasi-newton methods, an effective tool for approximating curvature in first-order smooth optimization, can be applied in the proximal setting, though at a significant price because their extensions to problems with additive structure may require a potentially costly computation of the scaled proximal operator. We show, however, that for quadratic support functions, we can exploit the simultaneous structure of f and g to compute the scaled proximal operator efficiently. When f and g have the right structure, the proximal map can be computed with cost nearly linear in the input size.

Finally, for problems involving large-scale statistical modeling, f is often a function written as expectation over a random variable, one with large, or potentially infinite support. In such a situation, even a single gradient computation is infeasible, requiring a costly integration. Stochastic approximation algorithms, however, demonstrate that it is unnecessary to compute the true gradient, and one only needs an unbiased estimator of it. It is known that by appropriately decreasing the variance of the error at each iteration, the expected rate of convergence matches that of the underlying deterministic gradient method. We show by appropriately decreasing an upper bound on the moment generating function, we have convergence with overwhelming probability. This guarantee rivals its deterministic counterpart.

In both these structures, first-order information is costly to compute but cheap, controllable approximations are available, thus reflecting the unifying theme of this work, “optimization with costly subgradients.”

Acknowledgments

This thesis was written in two universities — first, at the University of Britsh Columbia, and then at the University of Califonia, Davis. A constant through both, however, has been my supervisor Michael Friedlander. This thesis would not have been possible without his support and guidance, his attention to detail, his exacting standards and impeccable craftsmanship that extends both to papers and code. Thank you for making this possible.

My thanks also go to my dissertation committee, Jesus De Loera, Mattias Mkoeppe. Thank you for the detailed feedback, encouragement, and conversation. Thank you too, to Nick Harvey and Nando De Fredas for serving on my qualification committee, and being inspirational teachers in general.

Part 3 of this thesis was done in collaboration with my colleagues at Google, Andrew Cotter, and Maya Gupta. It is an immense pleasure working with you, and the ideas we've discussed continue to resonate with me to this day.

My thanks also go to the administrative staff of UBC, Joyce Poon, and UC Davis, Tina Denena and Sarah Driver. Thank you especially to Tina and Sarah for making the transfer smooth and for tolerating my occasional ignorance on matters administrative.

To my senior colleagues, Ting Kei Pong, Nathan Krislock, Ives Macedo and Ernie Esser (who passed away far too soon), my conversations with each one of you have shaped and sharpened my thinking about optimization immensely. You have been my unofficial mentors, and for your generosity of time and spirit, I thank you. And to my colleagues Will Wright, Ziyu Wang, Julie Nutini, Sasha Aravkin, and Rob Hocking, thank you for being excellent company and the delightful discussions. All of you have made this pursuit worthwhile.

To Shawn Carter and Chris Olah, it was a pleasure working with you on distill. Many of your innovative ideas on presentation and writing have found their way into this thesis. Thanks too, to Bill Aiello, Jason Swanson and Michael Todd for your valuable technical contributions.

Thanks to Marcos Ginestra, for the neverending support.

And to the many others I did not thank explicitly but helped me on this journey, thank you.

Research supported by ONR award N00014-17-1-2009.

Part 1

Introduction

CHAPTER 1

A tour of the thesis

1.1. Introduction

The applications of convex programming touch nearly every quantitative discipline, but in recent times there has been demand for fast, scalable algorithms for highly structured problems arising in the quickly growing field of machine learning. Machine learning is, to dramatically and unfairly simplify, the endeavor of fitting a model to data. This process of model fitting (sometimes referred to as training) involves a search over the model parameters to minimize misfit—a task usually phrased as an optimization problem. Convexity has planted deep roots in the field, which go back to the maximum likelihood formulations in exponential families [Bro86], maximum-margin classification [VK82], and boosting [FS95], and underlies many algorithms, especially in applications where fast, robust algorithms are required for training simple models on large data.

The advantages of convex programming in machine learning are twofold. First, convex optimization provides strong theoretical guarantees of optimality. These come in the form of optimality certificates and a rich collection of algorithms with convergence guarantees. These guarantees allow for a clean decoupling of the interests of modeling and optimization—it is the model, not the algorithm, that determines the solution to any model. Second, convexity provides a rich palette of functions for crafting models. These functions, in the right hands, allow the practitioner to exercise fine-grained control in the expression of beliefs about the data. The peculiarities of convex functions, such as non-smoothness, can be recast as a strength. The kinks and ridges of these functions can be engineered to encourage solutions with certain useful properties. We begin our thesis in Part 2 with a gentle introduction to convex modeling, to serve as a guide to the applications that drive the algorithms. We divide the problem of modeling into three parts: a loss, a regularizer and *dataset constraints*, the final of which is a novel framework published in Goh et al. [GCGF16].

The problems discussed in the previous section are, for the most part, fairly routine convex optimization programs and can be solved using standard tools. The demands of scale, however, require specialized algorithms for exploiting structure within a problem. We identify two classes of problems of interest—parametric optimization, and optimization with composite structure.

1.1.1. Notation. We use the symbol $\mathbf{1}$ to denote the constant vector of all ones. To isolate the i^{th} component of a vector we use the notation $[\cdot]_i$ to denote $\mathbf{1}_i^T x$ where $\mathbf{1}_i$ is the ith standard Euclidean vector. Horizontal vector concatenation is denoted, as is convention, with $[x_1, \cdots, x_n]$, but to avoid the need for cumbersome transposes, we borrow notation from MATLAB [MAT10] and let $[x_1; \cdots; x_n]$ denote vertical concatenation. To denote an interval, we bold brackets $\mathbf{[}a, b\mathbf{]}$–though the difference is subtle, the distinctions should be made clear from context. As is per convention, we use $\|\cdot\|_p$ to denote the p-norm, and set the default norm $\|\cdot\| := \|\cdot\|_2$. Quadratic norms, denoted $\|\cdot\|_H$ for H positive definite, are defined as $\|x\|_H = x^T H x$.

Let us now review some basic concepts in convex analysis. Consider a finite-dimensional convex function f takin values on the extended reals $\mathbf{R} \cup \{\infty\}$. We refer to the domain of the function f as all the points where it is real valued, $\operatorname{dom}(f) = \{x \mid f(x) < \infty\}$. In this thesis we only consider proper convex functions, functions where $\operatorname{dom}(f) \neq \emptyset$. The epigraph of a function is denoted by $\operatorname{epi}(f) = \{(t, x) \mid t \geq f(x)\}$, and the level set of a function is denoted by $\operatorname{lvl}_f(t) = \{x \mid f(x) = t\}$. The closure of a convex function, $\operatorname{cl}(f)$, is defined as the greatest lower semi-continuous function majorized by f. A closed convex function is a function where $\operatorname{cl}(f) = f$. Closed convex functions which are proper are exactly the set of lower semi-continuous functions [Roc70, Theorem 7.1], and thus we will refer to them interchangeably.

A subgradient of a function $f : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ at x is a element $g \in \mathbf{R}^n$ which proves a global affine minorant, i.e.,

$$f(\bar{x}) \geq f(x) + g^T(\bar{x} - x) \qquad \text{for all } \bar{x}. \tag{1.1.1}$$

The set of all vectors for which this is true defines the subdifferential $\partial f(x)$. When the subdifferential is a singleton, then the function is differentiable at that point and $\partial f(x) = \{\nabla f(x)\}$. A simple example of a non-differentiable convex function is the absolute value function, $f(x) = |x|$.

Though differentiable almost everywhere, it is not differentiable at 0, and it is easy to show that $\partial|\cdot|(0) = [-1, 1]$. For the n-dimensional 1-norm function, $f(x) = \|x\|_1$, the points of non-differentiability are the points where any coordinate is 0. These sets are of particular interest, as solutions of optimization problems tend to lie on these sets. Another useful extended real valued function is the convex indicator

$$\delta_C(x) \equiv \delta(x|C) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{if } x \notin C. \end{cases} \tag{1.1.2}$$

The convex indicator is used when enforcing a constraint, since $\inf_{x\in C} f(x) = \inf_x\{f(x) + \delta(x|C)\}$. Optimality in a convex function is easy to characterize. If $0 \in \partial f(x)$ then by definition

$$f(\bar{x}) \geq f(x) + 0^T(\bar{x} - x) \geq f(x) \qquad \forall \bar{x},$$

which implies immediately that x is a minimizer of f. The converse is true too, by observing that for any minimizer $\bar{x}$, $f(x) + 0^T(\bar{x} - x)$ is a lower minorant, and hence $0 \in \partial f(x)$.

A differentiable function has L-Lipschitz gradients if $\|\nabla f(y) - \nabla f(x)\| \leq L\,\|y - x\|$, and is strongly convex when $(\nabla f(x) - \nabla f(y))^T(x-y) \geq \mu\|x-y\|^2$, or equivalently when there exists a μ such that f admits a quadratic, rather than linear lower bound, $f(y) \geq f(x) + \nabla f(x)^T(\bar{x} - x) + \frac{\mu}{2}\|\bar{x} - x\|^2$ for all $\bar{x}$. When f is twice differentiable, these two properties allow the convenient characterization $\mu I \succeq \nabla^2 f(x) \succeq LI$, where $\succeq$ indicates an ordering with respect to the cone of positive semidefinite matrices.

1.2. Parametric optimization

The first class of problems we are interested in are problems defined implicitly over a subset of variables. For a real-valued function $h(\cdot,\cdot)$ jointly convex over its first and second arguments, these are problems where f is defined implicitly over a second *inner* optimization problem, i.e.,

$$\underset{x\in\mathcal{X}}{\text{minimize}} \qquad f(x) := \min_y h(x, y). \tag{1.2.1}$$

We assume $\mathcal{X}$ is a compact, convex set with a non-empty interior, and $h(x,\cdot)$ attains its minimum. A similar situation happens for a function $h(\cdot,\cdot)$ convex in the first argument, i.e.,

$$\underset{x\in\mathcal{X}}{\text{minimize}} \qquad f(x) := \max_y h(x,y). \tag{1.2.2}$$

We assume that the max is attained, and thus do not need to use sup. Problems of this kind emerge when there is useful structure in a subset of the variables (grouped here as y) that is not present jointly in (x,y). An example of this is the basis pursuit denoise problem. This is the problem of solving

$$\underset{y}{\text{minimize}} \qquad \|y\|_1 \qquad \text{s.t.} \quad \frac{1}{2}\|Ay-b\|^2 \le \epsilon.$$

In this situation it is sometimes convenient to solve the one dimensional dual,

$$\underset{x\ge 0}{\text{minimize}} \qquad \sup_y h(x,y) = \frac{x}{2}\left(\|Ay-b\|^2 - \epsilon\right) + \|y\|_1. \tag{1.2.3}$$

Each query for first-order information in (1.2.3) (a subgradient of the objective function) can be found by solving the the unconstrained inner problem

$$\underset{y}{\text{minimize}} \qquad \frac{x}{2}\|Ay-b\|^2 + \|y\|_1, \tag{1.2.4}$$

which admits efficient algorithms [VDBF08].

Since subgradients of f can be obtained, under mild assumptions (see Sections 5.1 and 5.2), by solving the subproblem of minimizing or maximizing h in its second variable and taking derivatives at the optimum, we can in principle apply any first-order method for optimizing f. Charging at this blindly with just any first-order method, however, is ill advised. First, each call for first-order information (such as a subgradient) is expensive, and we seek to use the most economical use of this information. Next, if evaluating f requires solving the inner problem numerically, intermediate steps of the optimization in h often yield partial solutions that can be integrated into the outer first-order algorithm.

The problem of efficient use of gradient information in convex optimization has been studied under the umbrella of *informational complexity* [Nem94]. An algorithm's information complexity, as the name suggests, is the number of subgradient calls required to drive down the objective value of a closed convex function below a certain tolerance [Nem94, Lecture 1]. Given how such calls to the subgradient are at a premium, it is natural for us to look for methods that are informationally efficient.

1.2.1. Efficient first-order algorithms. We begin our discussion with cutting plane methods, a class of methods capable of achieving optimal information efficiency. Newman [New65] motivates the development of cutting plane methods as a generalization of bisection. Recall that for a smooth, unimodal function, bisection works by progressively shrinking an interval P_k that contains the optimum by a factor of two. First, we choose the midpoint of the interval of P_k and query the function's gradient. Then, we rule out exactly half the interval by noting that a positive gradient at x implies the function is increasing, and hence the optimum lies to the left of x. A negative gradient implies the reverse—allowing us to cut P_k in half, increasing the precision of our estimate by exactly one bit each iteration.

When f is convex, a subgradient gives us similar localization information. Indeed, any linearization of f gives us an affine lower bound, and hence

$$f(\bar{x}) \geq f(x) + g^T(\bar{x} - x) \geq \min_x f(x) \quad \forall \bar{x} \text{ such that } g^T(\bar{x} - x) \geq 0.$$

Each query to the subgradient of f, thus, rules out a halfspace from containing the optimum, $\{x \mid g^T(\bar{x} - x) \geq 0\}$. Thus given any polytope P_0 which contains the optimum (this is known as the *localization polytope*), we can refine it by choosing a point in P_k and calling the subgradient oracle, and intersecting the halfspace with P_k, yielding the iteration

$$P_{k+1} = P_k \cap \{x \mid g^T(\bar{x} - x_k) \leq 0\}; \tag{1.2.5}$$

see Figure 1.2.1.

What remains undecided is what point to select. In one dimension the choice is obvious—the midpoint of an interval. But in higher dimensions, it is clear that a perfect generalization of the midpoint is impossible. There is no generic point in every convex set that neatly divides every half-space through that point into two equal parts. Methods must thus resort to some form of compromise. These are the methods of central selectors.

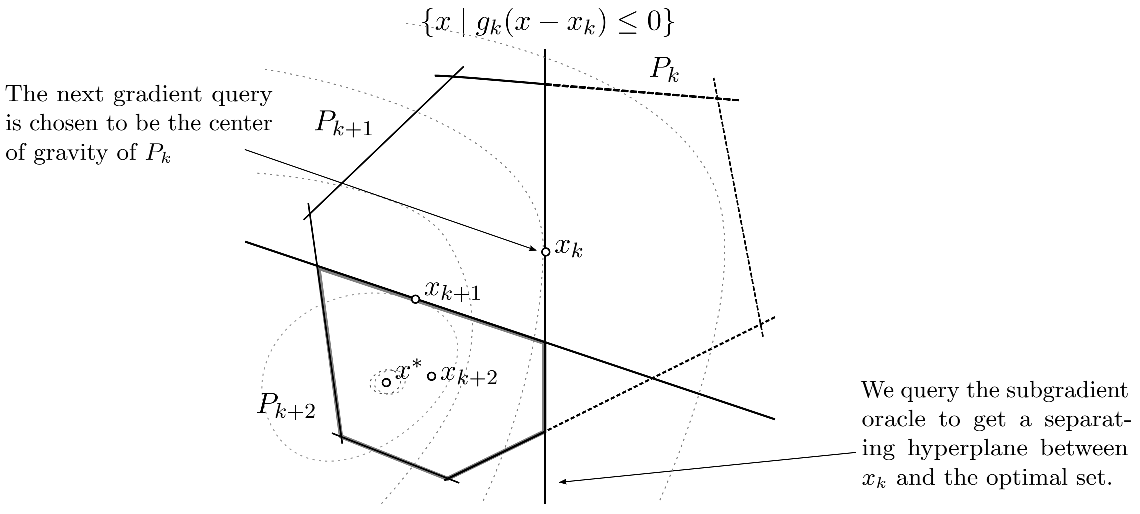

$\{x \mid g_k(x - x_k) \le 0\}$
P_k
The next gradient query is chosen to be the center of gravity of P_k
P_{k+1}
x_k
x_{k+1}
x^*
x_{k+2}
P_{k+2}
We query the subgradient oracle to get a separating hyperplane between x_k and the optimal set.

FIGURE 1.2.1. Graphical illustration of the cutting plane method

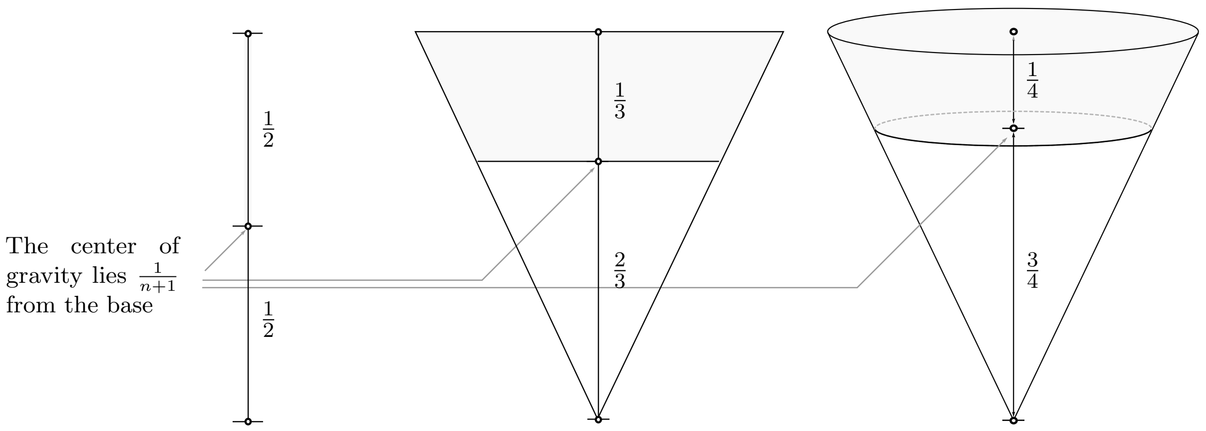


FIGURE 1.2.2. For a circular cone with n dimensions of height 1, the center of gravity lies at distance $1/(n+1)$ from the base, and the hyperplane $\{x \mid x_1 \le 1/(n+1)\}$ divides the cone into sets of volume $V(1+\frac{1}{n})^n$ and $V - V(1+\frac{1}{n})^n$, where V is the volume of the cone. As $n \to \infty$, the ratio of the volume of the shaded region to the cone approach $1 - \exp(-1)$, making the bound (1.2.6) asymptotically tight.

1.2.1.1. *Center of Gravity Method.* Levin [Lev65] and Newman [New65] suggested that the center of gravity would be a reasonable substitute for the midpoint. This choice takes advantage of Grünbaum's theorem [Grü60], a classical theorem in convex geometry. This theorem states that any hyperplane through the center of gravity necessarily splits the polytope "evenly", in the sense that if x is the center of gravity of P_k, the ratio of the volumes of P_k to $P_k \cap \{\bar{x} \mid g^T(x-\bar{x}) \leq 0\}$ will not exceed $1-\exp(-1)$ (see Figure 1.2.2 for an example where this bound is tight). Thus, rather than a decrease of a factor of 2 at every iteration, we settle for a factor of approximately 1.58—a sensible compromise [1]. With update (1.2.5) where x_k is the center of gravity of P_k,

$$\frac{\mathbf{vol}(P_{k+1})}{\mathbf{vol}(P_k)} \leq 1-\exp(-1). \tag{1.2.6}$$

This linear decrease in volume can be translated into a convergence rate to obtain a rate of convergence in the function value. The following result relies on Theorem 4.2.2, a reproduction of a classic proof adapted in Bubeck [Bub15].

Theorem 1.2.1. Consider the problem of minimizing a convex function f over a convex, compact set $\mathcal{X}$. Let $P_0 = \mathcal{X}$. Let $x_1, \ldots, x_k$ be the centers of gravity of $P_1, \ldots, P_k$, respectively. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \inf_x f(x) \leq (1-\exp(-1))^{\frac{k}{n}} \cdot \left(\max_{x\in\mathcal{X}} f(x) - \min_{x\in\mathcal{X}} f(x)\right)$$

A few remarks are in order. The center of gravity method enjoys universal, linear rate of convergence that is oblivious to the structure of the function itself. One can see this intuitively in the fact that the center of gravity is affine invariant, and hence the iterates are, up to a linear transform, equal for any affine scaling of f. Surprisingly, this method is optimal in the number of gradient queries, up to a constant factor, which makes it an excellent candidate for our problem [Nem94].

But this method is not without its shortcomings. First, this scheme degrades quickly in its effectiveness as the dimensions of the problem increase. The rate of convergence decreases exponentially as the dimensions increase, and thus the convergence rate slows down quickly for problems of any substantial dimension. Moreover, even for problems with modest dimension, the computation of the center of gravity quickly becomes an insurmountable task because the number of floating point operations (flops) needed to compute it is exponential in d [Nem94].

[1]We note in passing that the center of gravity of method can be applied to convex mixed-integer optimization under an appropriately defined measure, see Basu and Oertel [BO17].

1.2.1.2. *Ellipsoid Method.* The Ellipsoid method, first proposed by Shor, Yudin, A. S. Nemirovskii [BGT81] replaces the localization polytopes P_k with ellipsoids $\mathcal{E}_k$. This modification resolves several thorny issues present in the center of gravity method. First, the number of floating-points needed for representation is constant, $\mathcal{O}(n^2)$. Compare this, for example, to a polytope, which accumulates all information gathered so far. More importantly, there is little ambiguity concerning the choice of center—the center of the ellipsoid is a perfect generalization of the midpoint, for which every hyperplane divides $\mathcal{E}_k$ into two equal parts. With these observations, the ellipsoid method presents itself. Given an ellipse $\mathcal{E}_k$, we query the gradient oracle at its center and choose the next ellipsoid as

$$\mathcal{E}_{k+1} = \text{smallest ellipsoid containing } \mathcal{E}_k \cap \{x \mid g^T(\bar{x} - x) \leq 0\}.$$

It is clear that every ellipsoid in this sequence contains the optimum. But rather miraculously this update is sufficient to preserve linear convergence, as the ratio of volumes of the two ellipsoids can be shown never to exceed $\exp(-1/2n)$ [GLS12, Lemma 3.1.28].

Theorem 1.2.2. [Bub15, Theorem 2.4] Let $\mathcal{E}_0 \subseteq \mathcal{X}$ be a point which contains the optimum, and $x_1, \ldots, x_k$ be the centers of the spheres $\mathcal{E}_1, \ldots, \mathcal{E}_k$ respectively. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \inf_x f(x) \leq \exp\left(-\frac{1}{2n}\right)^{\frac{k}{n+1}} \cdot \left(\max_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} f(x)\right)$$

Indeed, this method, coupled with minor modifications to account for rounding errors, gave rise to Khachiyan's [Kha79] celebrated result resolving the then open question of the computational complexity of linear programming. Despite this, the rate of volume decrease is rather disappointing—the ellipsoid method has a doubly exponential dependence on dimension. And despite the initial optimism, and many refinements, using for example multiple or deep cuts [BGT81], it has not proved to be practical numerically.

1.2.1.3. *Bundle Methods.* Bundle methods take a different route altogether. These methods use the observation that we can "combine" any collection of lower affine minorants of the form $f(x_i) + g_i^T(\cdot - x_i)$ to form a lower model for f, and any collection of function values $f(x_i)$ to form

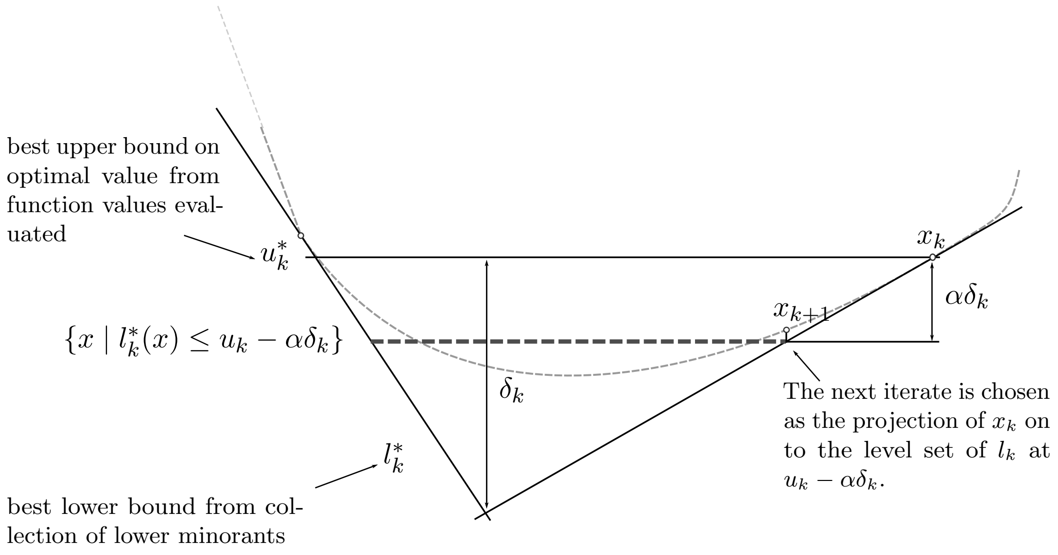

best upper bound on optimal value from function values evaluated
u_k^*
x_k
$\alpha\delta_k$
x_{k+1}
$\{x \mid l_k^*(x) \le u_k - \alpha\delta_k\}$
δ_k
The next iterate is chosen as the projection of x_k on to the level set of l_k at $u_k - \alpha\delta_k$.
l_k^*
best lower bound from collection of lower minorants

FIGURE 1.2.3. Graphical illustration of the level bundle method.

an upper bound on $\inf_x f(x)$. These bounds are

$$l_k^*(x) = \max_{0\le i<k} \{f(x_i) + g_i^T(x - x_i)\}, \qquad \text{the best lower bound on } f \text{ and} \tag{1.2.7}$$

$$u_k^* = \min\{f(x_0), \dots, f(x_{k-1})\}, \qquad \text{the best upper bound on optimal objective.} \tag{1.2.8}$$

Bundle methods then optimize the lower model in some way to find the next iterate. Its simplest variation is Kelly's method [Kel60], where this lower model is simply minimized at each iteration:

$$x_k = \operatorname*{argmin}_{x\in\mathcal{X}} l_k^*(x).$$

This iteration is valid due to the compactness of $\mathcal{X}$. This method is regarded as unstable and superseded by stabilized methods [LHU96], which add a quadratic term that encourages the next iterate to stay close to the current one, i.e.,

$$x_k = \operatorname*{argmin}_{x\in\mathcal{X}}\{l_k^*(x) + \frac{\alpha_k}{2}\|x - x_{k-1}\|^2\}.$$

The choice of α_k, which plays a similar role to a stepsize in descent methods, is chosen via a linesearch like method combining *serious* and *null* steps to ensure descent. The level bundle method

[LNN95] projects the next iterate onto some level set of the polyhedral lower bound,

$$x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \tfrac{1}{2}\|x - x_k\|^2 \mid \{x \mid l_k^*(x) \le u_k - \alpha\delta_k\} \right\}, \quad \text{where} \quad \delta_k = u_k - \min l_i(x), \tag{1.2.9}$$

where α is a number between 0 and 1. There are many more variations of this method [LNN95], and though they work well in practice, their theoretical guarantees are weak. Unlike the cutting plane methods, these algorithms do not succeed in obtaining a universal linear rate of convergence, only a sublinear one comparable to subgradient descent.

1.2.1.4. *The method of inscribed ellipsoids.* In a parallel development, Tarasov and Khachiyan [Tar88] returned to the method of central selectors, proposing a new choice of center. The center of the maximally inscribed ellipsoid (MIE) of P_k might serve as the generalization of the midpoint in lieu of the center of gravity. This method proved to be an excellent compromise. Via a subtle argument, Tarasov and Khachiyan showed that the following bound on the ratios of the volumes of the maximum inscribed ellipsoids of P_k,

$$\frac{\text{volume of maximum inscribed ellipsoid in } P_k}{\text{volume of maximum inscribed ellipsoid in } P_{k+1}} \le 0.843, \tag{1.2.10}$$

cf. Chapter 1.2.1.2. This theorem leads to a characterization of the convergence rate of the method of inscribed ellipsoids, stated in Theroem 4.2.2, reproduced below:

Theorem 1.2.3. Let $P_0 = \mathcal{X}$, and $x_1, \ldots, x_k$ be the centers of gravity of $P_1, \ldots, P_k$ respectively. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \inf_{x \in \mathcal{X}} f(x) \le 0.843^{\frac{k}{n+1}} \cdot \left(\max_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} f(x) \right).$$

Like the center of gravity method, the method of inscribed ellipsoids is affine invariant, and has optimal informational complexity, up to a constant factor. But the real value of this choice lies in the fact that the MIE is computable in polynomial time by solving a moderately sized semidefinite program. This puts the method of inscribed ellipsoids squarely in the purview of practical usage. Indeed, we find the method of inscribed ellipsoids to be very practical in situations when the cost of computing the maximally inscribed ellipsoid is dwarfed by the work required in computing the subgradient.

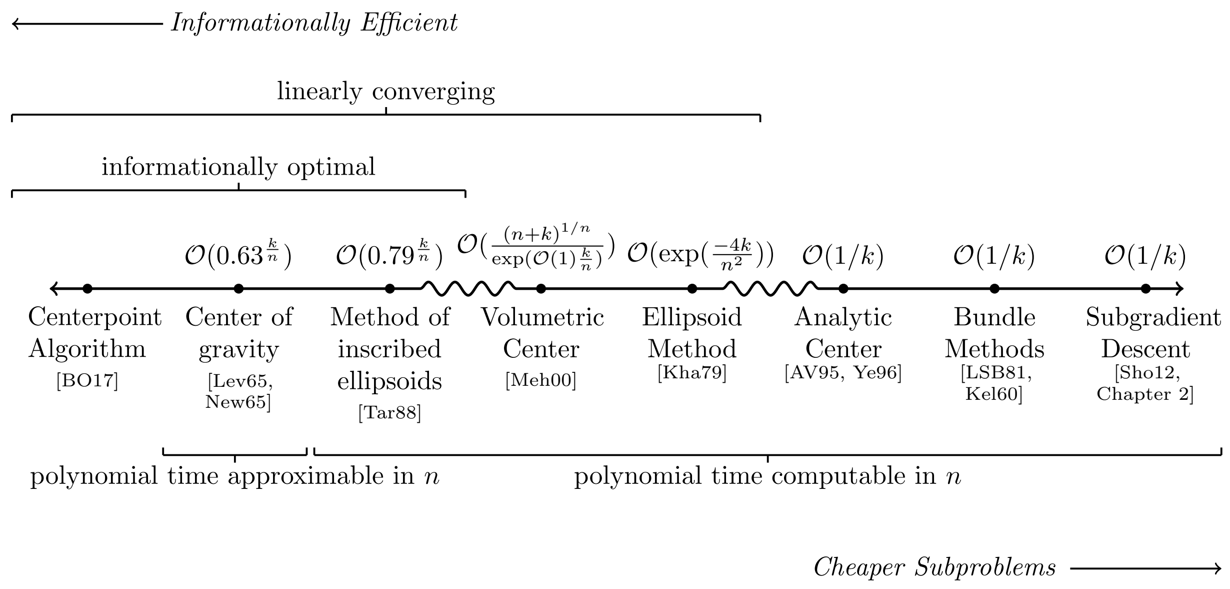

Informationally Efficient
linearly converging
informationally optimal
$\mathcal{O}(0.63^{\frac{k}{n}})$
$\mathcal{O}(0.79^{\frac{k}{n}})$
$\mathcal{O}(\frac{(n+k)^{1/n}}{\exp(\mathcal{O}(1)\frac{k}{n})})$
$\mathcal{O}(\exp(\frac{-4k}{n^2}))$
$\mathcal{O}(1/k)$
$\mathcal{O}(1/k)$
$\mathcal{O}(1/k)$
Centerpoint Algorithm [BO17]
Center of gravity [Lev65, New65]
Method of inscribed ellipsoids [Tar88]
Volumetric Center [Meh00]
Ellipsoid Method [Kha79]
Analytic Center [AV95, Ye96]
Bundle Methods [LSB81, Kel60]
Subgradient Descent [Sho12, Chapter 2]
polynomial time approximable in n
polynomial time computable in n
Cheaper Subproblems

FIGURE 1.2.4. Visual summary of convergence rates of first-order methods on general convex functions.

1.2.1.5. *Miscellanious Central Selectors.* We note two more choices of central selectors that have since been developed. First, the volumetric center [Vai89] comes within a *log* factor of being optimal and has the advantage that the center can be updated in $\mathcal{O}(n^3)$ arithmetic operations. Second, the analytic center can also be used [AV95, Ye96], with this choice permitting updates to the center in $\mathcal{O}(n)$ time. Unfortunately, this method is no longer affine invariant and can only be proven to have sublinear convergence. Despite this, there has been incredible success in the use of the analytic center method for solving large-scale, high dimensional optimization problems [BVS08, BGVDM94, Section 6]. We provide a visual summary of the methods discussed in this section in Figure 1.2.4.

1.2.2. Models with Inexact Subgradients. The above discussion assumes access to an oracle which returns a subgradient of f. This requires, in principle, an exact solution of the inner optimization problems of both (1.2.1) and (1.2.2). Such precision is often an overkill, however, and intermediate results of this inner optimization still yield useful information.

Consider problem (1.2.2) where $f(x) = \sup_y h(x,y)$. For a fixed vector x, any vector $\bar{y}$ in the domain of $h(x,\cdot)$ certifies a lower bound on f, viz. $h(x,\bar{y})$. Together with any subgradient

$g \in \partial h(\cdot, \bar{y})(x)$, this lower bound generates a global affine minorant

$$f \geq h(\cdot, \bar{y}) \geq h(x, \bar{y}) + g^T(x - \cdot).$$

Furthermore, if an upper bound u on $f(x)$ is known, then this yields the *inexact cutting plane*

$$\{\bar{x} \mid g^T(x - \bar{x}) \leq u - h(x, \bar{y})\}.$$

For both classes of problems (1.2.1) and (1.2.2), both upper bounds on $f(x)$ and lower affine minorants can be found through duality. We explore such constructions with examples in Chapter 5.

1.2.2.1. *Shallow Cuts.* The center of gravity discussed in Section 1.2.1.1 can be adapted for use with the inexact cutting planes. The key to doing so is a generalization of Grünbaum's theorem proved by Bertsimas and Vempala [BV04], presented here in a modified way. Define the covariance of a convex set **cov** to be

$$\mathbf{cov}(P) := \mathbf{E}[(z - \mathbf{E}z)(z - \mathbf{E}z)^T]$$

where z is a random variable distributed uniformly on P. This generalization then states that

$$\frac{\mathbf{vol}(P \cap \{\bar{x} \mid g^T(x - \bar{x})) \leq \epsilon \|g\|_{\mathbf{cov}(P)}\}}{\mathbf{vol}(P)} \leq 1 - \exp(-1) + \epsilon,$$

see Figure 1.2.5 for a worked example of this result. This motivates the definition of a shallow cut oracle [GLS12, Def 3.3.8].

Definition 1.2.4 (Shallow cut oracle Oracle)**.** Let f be a convex function. Then f is equipped with a shallow cut oracle if we can find for each (x, H, ϵ) a

$$g := \textbf{s-oracle}(x, H, \epsilon)$$

such that for all $\bar{x}$, $f(x) + g^T(\bar{x} - x) \leq \epsilon \|g\|_H$.

Since $\textbf{s-oracle}(x, H, 0) \in \partial f(x)$, it is clear that an algorithm which generates a sequence of vectors g_k converging to $g \in \partial f(x)$ will satisfy the shallow cut oracle condition eventually. The amount of work needed, however, is a complex interaction between g and H and is, in the absence of additional assumptions, difficult to bound a priori.

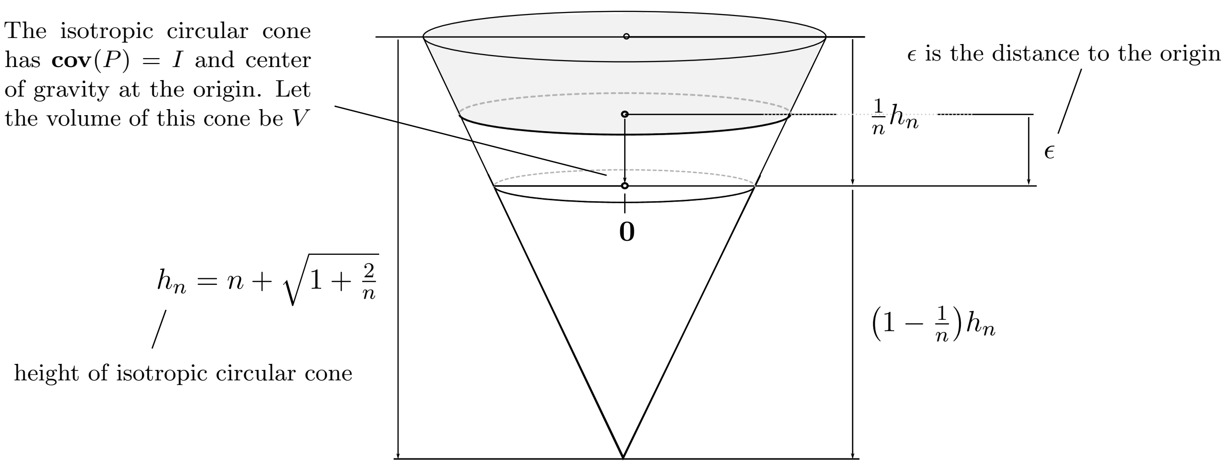

The isotropic circular cone has $\mathbf{cov}(P) = I$ and center of gravity at the origin. Let the volume of this cone be V
ϵ is the distance to the origin
$\frac{1}{n}h_n$
ϵ
$\mathbf{0}$
$h_n = n + \sqrt{1 + \frac{2}{n}}$
height of isotropic circular cone
$\left(1 - \frac{1}{n}\right)h_n$

FIGURE 1.2.5. Example of approximate error bound on the circular, isotropic cone. The volume of the unshaded region is $V(1 - \frac{1}{n} + \frac{\epsilon}{h_n})^n$ and the shaded region is $V - V(1 - \frac{1}{n} + \frac{\epsilon}{h_n})^n$. Thus the ratio of the volume of the shaded region to the whole cone is $1 - (1 - \frac{1}{n} + \frac{\epsilon}{h_n})^n \le 1 - \exp(\epsilon - 1) \le 1 - \exp(-1) + \epsilon$.

Assuming a shallow cut oracle is available, the center of gravity method can be modified trivially for use with shallow cuts. First choose a cutting plane, $g = \textbf{s-oracle}(x, \mathbf{cov}(P_k), \epsilon)$ for some $\epsilon < \exp(-1)$. Then updating the localization polytope according to

$$P_{k+1} = P_k \cap \{\bar{x} \mid g^T(x - \bar{x}) \le \epsilon \|g\|_{\mathbf{cov}(P_k)}\},$$

it is clear, by a similar line of reasoning to Theorem 4.2.2, that the shallow cut center of gravity method converges linearly, with a rate of $1 - \exp(-1) + \epsilon$.

Theorem 1.2.5. Let $P_0 = \mathcal{X}$. Let $x_1, \ldots, x_k$ be the centers of gravity of $P_1, \ldots, P_k$ respectively. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \inf_{x \in \mathcal{X}} f(x) \le (1 - \exp(-1) + \epsilon)^{k/(n+1)} \cdot \left(\max_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} f(x) \right).$$

A inexact version of the ellipsoid method exists. Let $\mathcal{E}_k = \{x \mid (x - \bar{x}_k)^T E_k^{-1}(x - \bar{x}_k) \le 1\}$. If we let $g_k = \textbf{s-oracle}(x, E_k, \epsilon)$, where $\epsilon < 1/n$ and

$$\mathcal{E}_{k+1} = \text{smallest ellipsoid containing } \mathcal{E}_k \cap \{x \mid g^T(\bar{x} - x) \le \epsilon \|g\|_{E_k}\}.$$

Then the shallow cut ellipsoid method guarantees linear convergence [GLS12]. This is the *shallow cut ellipsoid method*. The hit taken by using shallow cuts is encapsulated in the following theorem.

Theorem 1.2.6. [GLS12, Lemma 3.3.21] Let $\mathcal{E}_0 \subseteq \mathcal{X}$ be a point which contains the optimum, and $x_1, \ldots, x_k$ be the centers of the spheres $\mathcal{E}_1, \ldots, \mathcal{E}_k$ respectively. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \inf_{x \in \mathcal{X}} f(x) \leq \exp\left(-\frac{(1-n\epsilon)^2}{4n}\right)^{k/(n+1)} \cdot \left(\max_{x\in\mathcal{X}} f(x) - \min_{x\in\mathcal{X}} f(x)\right)$$

1.2.2.2. *Inexact Bundle Methods.* Bundle methods can also be modified to incorporate inexact information, though these methods require a different oracle to that in Definition 1.2.4. Here we measure the quality of a lower affine minorant, $l + g^T(\cdot - x)$ and an upper bound u on the optimum in a more straightforward way, simply by $u - l$ (see Figure 1.2.6). An inexact controllable oracle is one that demands a lower affine minorant in the form of a tuple (u, l, g) where $l - u$ does not exceed ϵ.

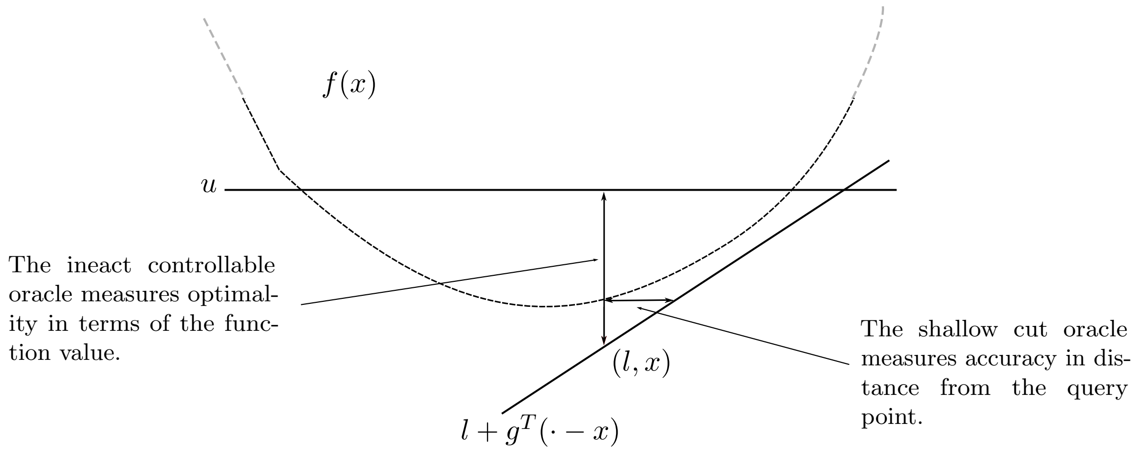

$f(x)$
u
The ineact controllable oracle measures optimality in terms of the function value.
(l, x)
The shallow cut oracle measures accuracy in distance from the query point.
$l + g^T(\cdot - x)$

FIGURE 1.2.6. Different ways of measuring optimality of an approximate subgradient.

Definition 1.2.7 (Inexact Controllable Oracle)**.** Let f be a convex function. Then f is equipped with an inexact controllable oracle if we can find, for a pair (x, ϵ) the tuple

$$(u, l, g) := \textbf{c-oracle}_f(x, \epsilon),$$

such that $l + g^T(\bar{x} - x) \leq f(\bar{x})$ for all $\bar{x}$, $f(x) \leq u$ and $u - l \leq \epsilon$.

The construction of such oracles are discussed in detail in Chapter 5. There we show that on many problems of interest, if we apply a primal-dual algorithm on the inner problem, we generate a sequence of upper bounds u_k, lower bounds $l_k + g_k^T(\cdot - x)$ for which

$$u_k \to f(x), \qquad l_k \to f(x), \qquad g_k \to g, \quad \text{where} \quad g \in \partial f(x).$$

Therefore, terminating this process at an appropriate point, we can obtain an affine lower minorant of any accuracy demanded by the inexact controllable oracle. We model the amount of work required to call $\textbf{c-oracle}_f(\cdot, \epsilon)$ as ϵ^{-1}, reflecting the fact that higher tolerances require greater computational effort (a more detailed justification of this model is given in Chapter 5).

Remark 1.2.1. If the upper bound is $u = f(x)$, then we can say for $(u, l, g) = \textbf{c-oracle}_f(x, \epsilon)$, g is in the familiar (see e.g. Claude and Lemaréchal [CL93, Equation 1.2]) ϵ-subgradient of f, written as $g \in \partial_\epsilon f(x)$.

In the inexact level bundle method, Fabian [Fáb00] observes that the level bundle method (1.2.9) can be adapted to incorporate these inexact lower minorants. For a choice of $0 < \gamma < (1-\alpha)^2$, where α is the level parameter set in equation (1.2.9), if we choose

$$(u_k, l_k, g_k) = \textbf{c-oracle}_f(x_k, \epsilon_k), \quad \text{where} \quad \epsilon_k = \gamma \cdot (u_k^* - \min_x l_k^*(x)) \tag{1.2.11}$$

and apply the updates

$$l_k^*(x) = \max_{i<k}\{l_k + g_i^T(x - x_i)\}, \qquad \text{in place of update (1.2.7) and} \tag{1.2.12}$$

$$u_k^* = \min\{u_1, \ldots, u_{k-1}\}, \qquad \text{in place of update (1.2.8),} \tag{1.2.13}$$

the convergence rate would only be impacted by a constant factor depending only on γ. This method, the *inexact level bundle method*, dynamically adjusts the error according to information available about the optimality of the problem, using rough estimates when far from optimality, and precise estimates as we approach it. This idea was subsequently refined by Oliveira and Sagastizábal [dOSL14] and Emiel and Sagastizabal [ES10].

1.2.3. The inexact epigraph cutting plane method. The method we propose draws inspiration from bundle methods and cutting plane methods, and attempts to synthesize the best features of both. The first observation comes from the fact that we can apply the cutting plane method to the epigraph of a problem, because the set

$$\operatorname{epi}(l_k^*) \cap \{(t,x) \mid t \le u_k^*\}$$

is a localizing polytope for *both* the optimal function value and an optimum, $(f(x^*), x^*)$. By choosing the next search point to be the center of this polytope, projected on to the search space, each query to an oracle now gives us two cutting planes,

$$\begin{aligned} \{(t,x) \mid t \le u_k\}, &\quad \text{an upper bound on } \inf_x f(x) \text{ and} \\ \{(t,x) \mid l_k + g_k^T(x - x_k) \le t\}, &\quad \text{a lower bound.} \end{aligned} \tag{1.2.14}$$

At least one of these cutting planes is guaranteed to cut off the center, and thus any of the choices of central selectors discussed above will guarantee the same decrease.

The epigraph cutting plane method that we describe above, the name of which we inherit from Boyd's lecture notes [BV07, Section 6], is not new. We have found scattered reference to this approach in the literature, with the earliest in [BGVDM94] and also [BV07, GV99, Meh00]. Our innovation lies in the observation that this method is particularly well suited to problems with inexact controllable oracles. Indeed, by switching our cutting plane to epigraph form, we can use the inexact controllable oracle 1.2.7, rather than the shallow cut oracle 1.2.4, thus allowing us to precisely bound the amount of work needed at each iteration. This forms the basis for the *inexact epigraph cutting plane method* described in Section 6.

In Section 7 we push this idea even further. We show that if f is strongly convex we can use parabolic lower bounds instead of linear lower bounds. This additional information ensures that the best lower bound on f is strongly convex at every iteration, and that this property prevents certain degeneracies of the localization polytope. This allows us to show that the sequence $\{\epsilon_k\}$ can be chosen in a geometrically decreasing series, allowing a bound on the total computational effort independent of the starting point.

1.3. Composite problems

We now consider problems of the form

$$\underset{x}{\text{minimize}} \qquad f(x) + g(x), \qquad \text{where} \qquad f(x) = \mathbf{E}_\xi[h(x,\xi)] = \int h(x,\xi)\, d\xi. \tag{1.3.1}$$

We assume $h(\cdot,\xi)$ is smooth, convex, with L-Lipschitz gradients for all ξ, and that the expectation is well defined for all x, and the minimum of $f+g$ is attained. The function g is possibly nonsmooth, but typically simple—a term that will be made concrete shortly. In machine learning applications, $h(\cdot,\xi)$ is typically a loss function, and the expectation is taken over the data generating process, where ξ is usually a feature-response pair.

1.3.1. Composite optimization. The properties of convexity and smoothness of $h(\cdot,\xi)$ are preserved in f, and thus f is smooth and convex. For simplicity, let us first consider the generic problem of a smooth function added to a convex function,

$$\underset{x}{\text{minimize}} \qquad f(x) + g(x). \tag{1.3.2}$$

Problems of this kind are studied under the umbrella of optimization with composite structure. No function composition is involved, however, and the name is a historical misnomer. The proximal update, a generalization of the method of steepest descent, is the tool of choice for exploiting this structure. The iteration can be interpreted as first constructing a local quadratic model of f at x_k,

$$f(x) + g(x) \approx \hat{f}_k(x) + g(x) \quad \text{where} \quad \hat{f}_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{\alpha_k}{2}\|x - x_k\|^2, \tag{1.3.3}$$

and then choosing the minimizer of that quadratic model, $\hat{f} + g$, as the next iterate. By completing the square we obtain the iterates for a sequence of stepsizes α_k. The iterates are defined by

$$x_{k+1} = \mathbf{prox}_{g/\alpha_k}(x_k - \alpha_k^{-1}\nabla f(x_k)),$$

where the proximal operator, a generalization of the projection, is defined as

$$\mathbf{prox}_g(x) = \underset{\bar{x}}{\text{argmin}} \left\{ \tfrac{1}{2}\|\bar{x} - x\|^2 + g(\bar{x}) \right\}. \tag{1.3.4}$$

When g is the indicator of a set, $\mathbf{prox}_{\delta_C}(x)$ is exactly the projection of x onto C. Since the minimizer is always unique, the proximal operator is well defined. When $\alpha_k \geq L$, where L is the Lipschitz coefficient of the gradients, $f \leq \hat{f}_k$, and the iteration is monotonic and converges.

The burden of computation for this iteration is spread over two operations—computing **prox** and evaluating the gradient. The former, though seemingly daunting, is often simple. When g is simple, $\mathbf{prox}_g$ can be evaluated in linear time, in closed form. For example, when g is the indicator on $[-1,1]^n$, the proximal operator is simply the projection onto a box, easily computable by doing a coordinate-wise clipping of x. And g, for most applications of interest in machine learning is simple—an extensive catalog of proximal operators is provided in the appendix of a survey by Combettes and Pesquet [CP11], which can be used out of the box.

For most problems of interest, therefore, the workhorse of computation lies in the evaluation of the gradient of f. For a generic ξ and h, this must be done numerically via Monte Carlo or quadrature, which is hard to compute to high accuracy, even for moderate dimensions. And even in situations where the formula can be computed in closed form, say, when ξ has finite support, this may still require a sum over a large dataset. It is thus imperative that we keep the gradient computations to a minimum. It is therefore of interest to understand the convergence rate of proximal gradient.

1.3.2. Preconditioned quasi newton. Let f be strongly convex with coefficient μ and with L-Lipschitz gradients, and let κ be the *condition number* of f, defined to be $\tau = \mu/L$. The convergence rate of proximal gradient with a fixed stepsize equal to $1/L$ can be shown to have linear convergence [LT93b]. We characterize the convergence rate in the following Lemma 11.1.1, reproduced below.

Theorem 1.3.1. Let

$$\pi_k := [f+g](x_k) - \min_x [f+g](x). \tag{1.3.5}$$

For the iteration (1.3.4),

$$\pi_k = \left(1 - \frac{1}{1+40\tau^2}\right)^k \pi_0.$$

A curious feature of the above theorem is that, unlike cutting plane methods, the convergence rate depends not on the dimension of f, but the conditioning of the problem, τ. The relationship between the condition number of f and convergence is no artifact of the proof—constructing failure modes for f is easy. Even for a two dimensional quadratic function, it is possible to construct simple numerical examples where a rescaling of the variables cause the iterations to slow down arbitrarily. We thus seek a way to reduce the number of oracle calls to $\nabla f(x)$.

The scaled proximal-gradient method attempts to address ill-conditioning at its root. The strategy proceeds by using a different quadratic approximation of curvature in (1.3.3),

$$f(x) \approx f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}\|x - x_k\|_H^2 + g(x)$$

where H is an approximation of the curvature of the problem. This results in a scaled proximal iteration,

$$x_{k+1} = \mathbf{prox}_g^{H_k}(x_k - H_k^{-1}\nabla f(x_k)),$$

where we define the *scaled proximal operator* as

$$\mathbf{prox}_g^H(x) = \underset{\bar{x}}{\operatorname{argmin}} \left\{ \tfrac{1}{2}\|\bar{x} - x\|_H^2 + g(\bar{x}) \right\}. \tag{1.3.6}$$

To understand why this works, let us consider the simple example where $H_k = H$ for a symmetric and positive definite matrix H, where f is twice differentiable. We can then see the scaled proximal gradient iteration as proximal gradient on Problem 1.3.2 with the change of variables $y = H^{1/2}x$. The new condition number is now derived from uniform bounds on the symmetrically preconditioned Hessian

$$H^{1/2}\nabla^2 f(x) H^{1/2}.$$

The preconditioned matrix has the potential to be dramatically better conditioned. In the extreme case where $f(x) = \frac{1}{2}x^T Hx$, the scaled proximal gradient algorithm trivially converges in a single step.

Of course, we cannot ignore the sleight of hand involved here. The careful reader may have observed we have merely shifted the work from one place to another. Though we may have reduced the number of gradient calls needed, we must now evaluate the scaled proximal operator (1.3.6).

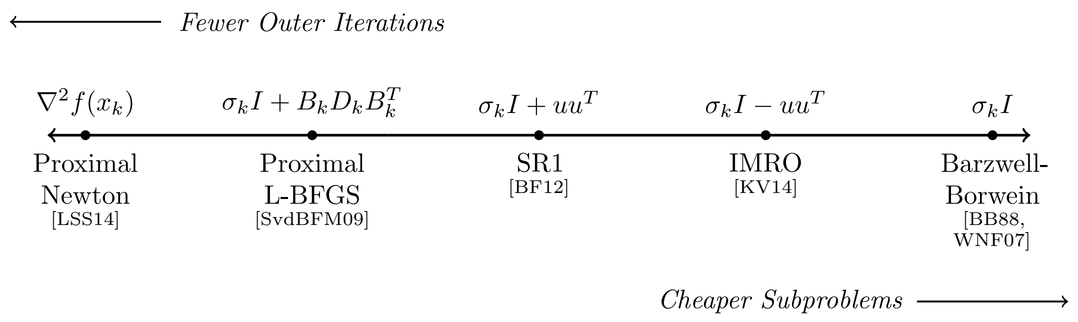


FIGURE 1.3.1. A summary of various proximal quasi-newton approximations in the literature

In the case where $g = \delta(\cdot|[-1,1]^n)$, for example, for a generic H, computing the scaled proximal operator (1.3.6) is a generic bound constrained quadratic program.

The choice of H remains very much an art. There is, by virtue of the freedom in selecting H, an entire continuum of algorithms that can be designed, depending on the intricacy of the curvature approximation. At one extreme, if f is twice differentiable, we can use the Hessian matrix directly. This is the proximal Newton method [LSS14], which admits super-linear convergence. At the other extreme, we have H as a multiple of the identity, which reduces to proximal gradient, perhaps with a carefully chosen steplength [BB88, WNF07]. The trick to an effective scaling is to find the right balance of structure and approximability—simple, structured matrices H that also approximate the Hessian well.

1.3.2.1. *Proximal Quasi-Newton.* We turn to a class of quasi-Newton methods that generate matrices which approximate second order information from first-order information, effectively approximating curvature on the fly. We follow Nocedal and Wright [NW99, §6.1], who use H_k to denote the current approximation to the *inverse* of the Hessian of f. Let x_k and x_{k-1} be two consecutive iterates, and define the vectors

$$s_k = x_{k+1} - x_k, \quad \text{and} \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

Here we give a summary of a few popular quasi-Newton methods used to obtain curvature approximations of a smooth function f.

Symmetric Rank-1 (SR1): The SR1 update uses the recursion

$$H_0 = \sigma I, \qquad H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k},$$

for some positive parameter σ that defines the initial approximation. For numerical stability, we skip the update (i.e. $H_{k+1} = H_k$) when $|(s_k - H_k y_k)^T y_k| \geq r\|s_k\|\|s_k - H_k y_k\|$.

Symmetric Rank-0 (SR0): The SR0 update uses a single update in the SR1 recursion, where σ is chosen carefully. The curvature approximation is

$$\bar{H} = \frac{s_k^T y_k}{\|y_k\|^2} I, \qquad H_k = \bar{H} + \frac{(s_k - \bar{H} y_k)(s_k - \bar{H} y_k)^T}{(s_k - \bar{H} y_k)^T (y_k)}$$

if $(s_k - \bar{H} y_k)^T y_k \geq \epsilon \|y_k\|\|s_k - H_0 y_k\|$, and $H_k = \bar{H}$ otherwise.

BFGS: The BFGS method updates the approximation H_k via the recursion

$$H_0 = \sigma I, \qquad H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

for some positive parameter σ that defines the initial approximation.

L-BFGS: The limited-memory variant of the BFGS update (L-BFGS) maintains the most recent m pairs (s_k, y_k), discarding older vectors. In all cases, $m \ll n$, e.g., $m = 10$. This ensures that H_k permits the representation

$$H_k = \sigma_k I + B_k D_k B_k^T,$$

where B_k is a $n \times 2m$ matrix and D_k is a $2m \times 2m$ matrix.

Global quasi-Newton: The globalization strategy advocated by Scheinberg and Tang [ST16] may add a small multiple of the identity to H_k. This modification takes the place of a potentially expensive linesearch, and the correction is increased at each iteration if a certain condition for decrease is not satisfied.

These methods have its origin in smooth optimization [NW99], but there has been considerable work adapting these curvature approximations to proximal gradient. The primary challenge in this approach is to effectively exploit the simultaneous structure of H and g to efficiently compute (1.3.6) to high precision. The first generation of techniques uses generic optimization tools to compute the scaled proximal operator. Schmidt et al. [SvdBFM09] computes $\mathbf{prox}_g^H$ using proximal gradient, and demonstrates extremely promising improvements in the number of required calls to ∇f. But the use of proximal gradient for the subproblems is not without flaws: many flops are required to compute the scaled proximal operator to high precision, which is exacerbated when H is ill conditioned—exactly the kinds of matrices H that make good models of curvature when f is ill conditioned. Other proposals to compute the scaled proximal operator include coordinate descent, when g is separable, or active-set Newton [BCNO12], when $g = \|x\|_1$. The latter, which may achieve high accuracy, is not guaranteed to converge, and may cycle [BCNO12, Chapter 4.1]. Furthermore, the algorithm becomes very intricate for any function g which is not separable.

Becker and Fazeli [BF12] made a first attempt in this direction for matrices H of the form $H = \Sigma + aa^T$, where Σ is a diagonal matrix and a is a vector. The scaled proximal operator was solved by dualizing to obtain a one-dimensional problem, and subsequently solving that via bisection—a generalization of the Sherman-Morrison formula. This result applies to the $SR0$ and performs admirably in practice, but is severely restrictive in the choice of quasi-Newton approximations. For example, even the L-BFGS matrix with memory 1 is not covered in this framework. A very similar structure is exploited in computing the proximal operator for matrices of the form $H = \Sigma - aa^T$ [KV14], but this technique has similar limitations.

In summary, what we require is a method for computing the scaled proximal operator that is highly precise, and allows us to take advantage of any structure in H and g insofar as they might exist. We propose such a technique in Chapter 1.3.6, published in Friedlander et al. [FG17]. Central to our method is the observation that many functions g of interest fall into a class of piecewise linear quadratic functions, a generalization of the sub-linear support function [Roc70, Chapter 13] and a modification of the piecewise linear quadratic functions [ABP13].

The advantages of this method are not theoretical. This algorithm forms the basis for the solver `QSip`, a quasi-newton L-BFGS proximal solver. In Section 9.6, we subject our solver to a battery of tests and show that we can make considerable gains in wall-clock time in solving dense problems with composite structure.

1.3.3. Stochastic gradient with error. While reducing the number of calls to ∇f provides some welcome reprieve, there are many situations where even a single evaluation of ∇f is beyond our computational reach. Recall that in (1.3.1) f can be written as

$$f(x) = \mathbf{E}_\xi[h(x,\xi)] = \int h(x,\xi)\,d\xi.$$

This expectation may be taken, for example, over an infinite stream of data, say one produced from a natural source such as the internet. Barring waiting for an inordinate amount of time to collect enough samples, it might be beneficial to use an approximation with the samples already collected. Assuming we have, for simplicity, an independent and identically distributed stream of random variables ξ_i, taken from the same distribution as ξ, one possible approximation averages $\nabla h(\cdot,\xi)$ over the samples, i.e.,

$$g_k = \frac{1}{m}\sum_{i=0}^{m} \nabla_x h(x_k,\xi_i). \tag{1.3.7}$$

Because g_k is an unbiased estimator of $\nabla f(x)$, it seems reasonable for one to use g as a drop in replacement for $\nabla f(x)$:

$$x_{k+1} = \mathbf{prox}_g^{H_k}(x_k - H_k^{-1} g_k). \tag{1.3.8}$$

It is thus convenient to think of this as proximal gradient with error, because

$$g_k = \nabla f(x_k) + e_k \qquad \text{where} \qquad e_k = \frac{1}{m}\sum_{i=0}^{m} \nabla_x h(x_k,\xi_i) - \mathbf{E}[\nabla_x h(x_k,\xi_i)].$$

In (1.3.7), g_k is an example of a gradient approximation coming from a *stochastic oracle*—which we define as any unbiased estimator of the gradient with finite variance. The quality of a stochastic oracle depends on the magnitude of $\|e_k\|$, and we can measure this in three different ways. In Section 10, we introduce three different stochastic oracles, A-**sto-oracle**, which measures the error deterministically, B-**sto-oracle**, which measures the error in terms of high probability bounds, and C-**sto-oracle**, which measures the error in expectation. More detailed discussion of stochastic oracles, and their construction, can be found in Section 10.

With the introduction of the Monte-Carlo estimator g_k, our iterates are now random variables. We consider the default mode of convergence to be convergence in probability:

$$\lim_{k\to\infty} \Pr(\pi_k \geq \epsilon) = 0 \qquad \text{for all } \epsilon.$$

Convergence rates, where shown, are typically done by bounding the expectation $\mathbf{E}[\pi_k]$ though we have more to say on this issue shortly. Note that by the non-negativity of π_k and Markov's inequality, convergence in expectation also implies convergence in probability.

1.3.3.1. *Stochastic Approximation.* Let us consider, for simplicity, the unscaled proximal iteration where $H_k = \frac{1}{L} I$. The simplest method we can think of uses the iteration (1.3.8) with no modification, and is based on using the same number of samples at every iteration. Unfortunately, this naive approach fails to converge. The expected rate of function decrease can be proven to be, for π_k defined in equation (1.3.5)

$$\mathbf{E}\pi_k \leq \rho^k \pi_0 + \frac{1}{\vartheta} \sum_{i=0}^{k-1} \rho^{k-1-i} \mathbf{E}\|e_i\|^2,$$

where

$$\rho = 1 - \frac{1}{1+40\tau^2} \in (0,1) \qquad \text{and} \qquad \vartheta = L \cdot \left(\frac{1}{40\tau^2} + 1 \right) > 0,$$

where τ is the condition number of the problem. This result is described fully in Lemma 11.1.1 of Section 11.1—and this expression is telling. The first term on the right-hand side of the inequality decreases at a linear rate, and depends on the condition number through ρ. This term is also present for any deterministic first-order method with constant step size. The second is an additive quantity depending on the level of the error. This suggests that in the naive approach to proximal gradient descent, each run can be broken into two phases. First, a "transient phase", where the decrease is dominated by ρ, and progress rapid. And second, a "fine-tuning" phase, where the optimization stalls. The goal, thus, of a converging stochastic gradient algorithm is to suppress this latter, fine-tuning phase, so that steady convergence is achieved.

Decreasing Step Size. One way to control the accumulation of error is to decrease the step size. This approach dates back to the classical results of Robbins and Monoro [RM51], and suggest a schedule of step sizes going down according to a divergent-series rule

$$\alpha_k > 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty, \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

The harmonic sequence $\alpha_k = \mathcal{O}(1)/k$, for example, satisfies this property. Under mild conditions it can be shown that with this schedule, proximal gradient convergences in probability to the optimum [NB00, Proposition 3.2] with probability 1. With strong convexity and a very carefully chosen step size, Nemirovski [NJLS09] proves a convergence rate of $\mathcal{O}(1/k)$.

Iterate Averaging. The convergence rate shown in the Nemirovski's proof [NJLS09], however, very brittle, and requires a precise choice of step size based on the coefficient of strong convexity of f. Stochastic averaging sacrifices some convergence speed for robustness, and uses the same iterative scheme (1.3.8) with a *constant* step size but suggests a twist: the output of the algorithm is the average of the iterates,

$$\bar{x}_k = \frac{1}{k} \sum_{i=1}^{k} x_i,$$

not the x_k's themselves. Though the iterates themselves do not converge, the average then can be shown to converge at a slightly slower rate $\mathcal{O}(1/\sqrt{k})$. This averaging scheme, though slower, does not require a precise choice of the step size and thus is termed *robust stochastic approximation.* Note that the rate of $\mathcal{O}(1/k)$ can be restored in if the errors are summable, $\sum_{i=1}^{\infty} \|e_k\|^2 < \infty$, even in the absence of strong convexity [SRB11, Proposition 1]. A similar result is proved by Friedlander and Schmidt [FS12].

Dynamic batching. The third strategy, *dynamic batching*, changes the sample size while keeping the step size constant. Luo and Tseng [LT93b] show that under certain error conditions, when g is an indicator and for a decreasing $\|e_k\| = \mathcal{O}(\|x_{k+1} - x_k\|)$, the function values converge to the optimal value linearly. Friedlander and Schmidt [FS12] suggest decreasing the errors according to explicit schedule, determined by the condition number of the problem, and show that linear convergence can be achieved by decreasing the errors at the rate

$$\begin{aligned} \|e_k\|^2 &\le \tau\beta^k \quad \text{in the deterministic case and} \\ \mathbf{E}\|e_k\|^2 &\le \tau\beta^k \quad \text{in the stochastic case.} \end{aligned} \tag{1.3.9}$$

for some τ and β. A suggestive result from Byrd [BCNW12] shows that in limited situations, such a schedule of exponentially increasing batch sizes in fact improves the total *sample complexity*, the total number of independent samples needed to achieve optimality of a certain tolerance over methods involving a decreasing stepsize methods and iterate averaging.

1.3.4. Tail bounds. All the convergence rates discussed above discuss convergence in expectation. This, in principle, tells us how the convergence of the algorithm fares over repeated runs, and says less about the concentration of our estimates, or how tightly these runs are grouped about the mean. One may instead be interested in confidence level solutions, as proposed by Nesterov [NV08]:

$$\Pr(\pi_k \geq \epsilon) \leq 1 - \beta,$$

for some $\epsilon > 0$ and $0 < \beta < 1$ with π_k defined in (1.3.5). Of course, one can trivially get such a bound via Markov's inequality, to deduce

$$\Pr(\pi_k \geq \epsilon) \leq \frac{\mathbf{E}\pi_k}{\epsilon}.$$

But we can do much better with a more refined analysis. Indeed, it is often the case that under very light assumptions on the distribution we are sampling, it can be shown that the iterates concentrate exponentially around the mean. Nemirovski, Juditsky, Lan and Shapiro [NJLS09] provides such an analysis show that for decreasing stepsizes $\alpha_k = \mathcal{O}(1/k)$, and Nesterov [NV08, Theorem 3] proves such confidence level bounds in the context of iterate averaging.

In Chapter 11, we provide a rigorous analysis of the tail bounds for the generic proximal gradient algorithm with error with a constant step size. The primary observation here connects the sub-gaussianity of the components of e_k,

$$\Pr\left([e_k]_i \geq \delta\right) \leq \exp\left(-\delta^2/\epsilon\right)$$

to the concentration around the mean in the iterates. We give tools for translating bounds on the moment generating function of $\|e_k\|^2$ to tail bounds on π_k, with an application to batching with exponentially increasing batchsizes [FS12]. The work in this chapter has been compiled in the paper [FG13].

1.4. The problem of costly subgradients

A common ingredient in both these structures is the scarcity of first-order information and the availability of cheap, controllable approximations to the gradient that can be made arbitrarily accurate. Thus the title of the thesis, “optimization with costly subgradients”, reflects the unifying theme of this work.

CHAPTER 2

Convex duality

Duality is the study of the pairing of convex programs, a *primal* and a *dual*, where the properties of one can be related to the other. The exercise of deriving an optimization program's dual, though largely a mechanical process, is often an instructive one. The dual often provides fresh perspective on the problem's structure, properties, and gives new insight into ways of tackling it.

The theory of convex duality is a generalization of the theory of linear programming duality [GKT51], and in this presentation we favor the duality framework advocated by Fenchel [ET99] and Rockafellar [Roc64, R$^+$66, Roc67] that relies heavily on the concept of convex conjugacy [Fen49]. This current presentation does not attempt to provide a comprehensive view of the theory of convex duality, but serves merely as a gentle introduction to the concepts and notation used in this thesis. We refer the reader to Rockafellar and Wets [RW98] for a through treatment of these topics.

A familiar example of convex duality, the Pythagorean equation, serves as a starting point for the oncoming discussion. Given two orthogonal subspaces A and $A^\perp$, and a point b, the Pythagorean equation states that

$$\min_x\{\tfrac{1}{2}\|x-b\|^2+\delta(x|A)\}+\min_v\{\tfrac{1}{2}\|v-b\|^2+\delta(y|A^\perp)\}=\tfrac{1}{2}\|b\|^2. \tag{2.0.1}$$

The subspaces A and $A^\perp$ are examples of orthogonal *dual cones*. The functions $\delta(x|A)$ and $\delta(v|A^\perp)$ are *conjugate* to each other, and the respective projections are dual convex programs. We explore each of these dualities in order.

2.1. Polar cones

Any nonzero vector in $\mathbf{R}^n$ can be decomposed into two components, its magnitude $\|x\|$ and its direction $x/\|x\|$. A convex cone is a set of directions, or *rays* in $\mathbf{R}^n$. Formally, the set $\mathcal{K}$ is a convex cone if $\mathcal{K}$ is a convex set and for every $\alpha > 0$, $\alpha x \in \mathcal{K}$. For any cone, we can define its polar to be

$$\mathcal{K}^\circ = \{v \mid x^T v \leq 0, x \in \mathcal{K}, v \in \mathbf{R}^n\},$$

and the dual cone $\mathcal{K}^* = -\mathcal{K}^\circ$. In general $\mathcal{K}^{**} = \text{cl}(\text{conv}(\mathcal{K}))$ [Roc70, Theorem 14.1], where $\text{conv}(\cdot)$ is the convex hull of $\mathcal{K}$. When a nonempty cone is closed and convex, it is clear that

$$\mathcal{K}^{\circ\circ} = \mathcal{K}, \qquad \mathcal{K}^{**} = \mathcal{K}.$$

Every convex cone defines a partial ordering of the vectors in $\mathbf{R}^n$, i.e.

$$x \preceq_{\mathcal{K}} y \iff x - y \in \mathcal{K}, \tag{2.1.1}$$

see [RW98, Proposition 3.38]. This property serves as the basis of *conic programming*, a generalization of linear programming, where the linear inequality constraints are replaced with conic constraints of the form $Ax \succeq_{\mathcal{K}} b$. These conic constraints, far from a mathematical curiosity, have been found to be of immense practical value in the modeling of real world problems. Here we consider several examples of convex cones.

Subspaces: $\mathbf{R}^n$ is a cone, with dual and polar equal to trivial set $\{0\}$. More generally, all subspaces are cones, and their polars are their orthogonal complements. The subspace $\{0\}$ induces the trivial ordering $x \preceq_{\{0\}} y \iff x = y$.

Polyhedral Cones: Polyhedral cones are cones of the form $\mathcal{K} = \{x \mid Bx \leq 0\}$ for some matrix B. The polar is $U^\circ = \{B^T y \mid y \leq 0\}$. The nonnegative orthant, $\mathbf{R}_+$, for example is a convex cone, with polar equal to the negative orthant $\mathbf{R}_-$. $\mathbf{R}_+$ induces the ordering $x \preceq y$ if and only if $x \leq y$ componentwise, which is the foundation of linear programming. Another example of a polyhedral cone is the isotonic cone $\mathcal{K} = \{x \mid x_1 \leq \cdots \leq x_n\}$. The ordering induced by the isotonic cone is $x \preceq y$ when the differences between successive elements are elementwise positive.

Second-Order Cone: The second-order cone is the set

$$\mathcal{Q}^m := \{(\tau, z) \in \mathbf{R} \times \mathbf{R}^{m-1} | z \in \tau \mathbf{B}_2\} \qquad \text{where} \qquad \mathbf{B}_p = \{z \mid \|z\|_p \leq 1\}. \tag{2.1.2}$$

It is self dual, i.e., $(\mathcal{Q}^m)^* = \mathcal{Q}^m$. The order induced by the second-order cone can be characterized by the ordering of positive semidefinite matrices induced by the arrow matrix, i.e.

$$x \preceq y \iff \mathbf{arrow}(x) \preceq \mathbf{arrow}(x)$$

where

$$\mathbf{arrow}(x) := \begin{pmatrix} x_0 & \bar{x}^T \\ \bar{x} & u_0 I \end{pmatrix} \quad \text{for} \quad x = [x, \bar{x}]. \tag{2.1.3}$$

The relationship between a cone and its polar satify a number of pleasing properties. Let $\mathcal{K}_1$ and $\mathcal{K}_1$ be nonempty closed and convex cones. Then the dual cone reverses inclusion, i.e. $\mathcal{K}_1 \subset \mathcal{K}_2$ implies $\mathcal{K}_2^\circ \subset \mathcal{K}_2^\circ$. Furthermore, for the Minkowski products and sum

$$\lambda \mathcal{X} = \{\lambda x \mid x \in \mathcal{X}\}, \qquad \mathcal{X}_1 + \mathcal{X}_2 = \{x_1 + x_2 \mid x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\}, \tag{2.1.4}$$

it is easy to show $\lambda \mathcal{K} = \mathcal{K}$ for $\lambda > 0$ and $(\mathcal{K}_1 + \mathcal{K}_2)^\circ = \mathcal{K}_1^\circ \cap \mathcal{K}_2^\circ$, generalizing the familiar subspace inequality that $(A + B)^\perp = A^\perp \cap B^\perp$.

2.2. Conjugate functions

The Legendre-Fenchel transform is mapping of functions to functions $f : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ to functions $f^* : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ where

$$f^*(v) = \sup_x \{v^T x - f(x)\}.$$

The function f^*, called the *conjugate* of f, can be interpreted as a bookkeeping tool that indexes all affine functions majorized by f. Indeed, any point $(t, g) \in \text{epi}(f^*)$ is an affine minorant of f of the form $x \mapsto g^T x - t$ [Roc70, Page 104].

The conjugate of any function f is convex (this can be seen from the fact that it is a supremeum of linear functions), and the bi-conjugate f^{**} fills in all the nonconvex gaps of f unreachable by any affine minorant, i.e. $f^{**} = \text{cl}(\text{conv}(f))$, where $\text{conv}(f)$ is the convex envelope of a f [RW98, Theorem 11.1]. When f is closed and convex, we get the pleasing duality

$$f^{**} = f.$$

Since the subgradients of convex functions are, by the subgradient inequality, also lower affine minorants, it stands to reason this conjugacy also gives us local variational information about f. This is indeed the case—for smooth f with an invertable gradient map (e.g. if f is strongly convex), $(\nabla f)^{-1} = \nabla f^*$. For gradient maps which aren't single valued or invertable, a more sophisticated version of this theorem exists, which states that in general $v \in \partial f(x) \iff x \in \partial f^*(v)$ [RW98, Proposition 11.3]. We describe several important examples of conjugate functions:

Linear Functions: The conjugate of a linear function's support is the singleton equal to the slope of the function, $f(x) = a^T x, f^*(x) = \delta(x|\{a\})$. The conjugate of the 0 function, in particular, is the indicator on 0.

Quadratics: When Q is symmetric, and positive definite, the conjugate of a quadratic function is the quadratic of its inverse,

$$f(x) = \frac{1}{2}x^T Q x, \qquad f^*(x) = \frac{1}{2}x^T Q^{-1} x.$$

The special case of $f(x) = \frac{1}{2}x^T x$ is the unique self dual function.

Support Functions: Support functions are conjugates of convex indicators

$$f(x) = \delta(\cdot|C)^*(x) = \sup_{v \in C}\{v^T x\},$$

and are useful for modeling penalties as they are positively homogeneous, $f(\lambda x) = \lambda f(x)$. We note two special cases. The conjugate of an indicator on the cone is the indicator on the polar cone: $\delta(\cdot|\mathcal{K})^* = \delta(\cdot|\mathcal{K}^\circ)$, and any norm ($\|\cdot\|$) is the support function of the dual norm ($\|\cdot\|_*$) ball

$$\|x\| = \delta(\cdot|\{x \mid \|x\|_* \leq 1\})^*(x) = \sup_{\|v\|_* \leq 1} v^T x.$$

The conjugate of the max function $f(x) = \max\{x_1, \dots, x_n\}$ is the support function of the standard simplex $\{x \mid \mathbf{1}^T x = 1, x \geq 0\}$.

Logs and Exponentials: The conjugates of the log and exponential functions are

$$\begin{aligned} f(x) &= \exp(x), & f^*(v) &= v\log(-v) + v, \\ f(x) &= \log(x), & f^*(v) &= -1 - \log v, \end{aligned}$$

see [Roc74, Eq 3.19]. The conjugate of the entropy function is the softmax function

$$f(x) = \log\left(\sum \exp(x_i)\right), \qquad f^*(v) = \begin{cases} \sum v_i \log v_i & \text{if } \mathbf{1}^T v = 1, v \geq 0 \\ \infty & \text{otherwise.} \end{cases}$$

Where we use the convention $0 \log 0 = 0$. Finally the conjugate of the logistic function is

$$f(x) = \log(1 + \exp(-x)), \qquad f^*(v) = \begin{cases} v\log(v) + (1-v)\log(1-v) & \text{if } v \in [0,1] \\ \infty & \text{if } v \notin [0,1]. \end{cases}$$

We list a few more properties of convex functions of interest. Trivially, from the definition of a convex conjugate we have $f^*(x) + f(v) \geq x^T v$, the Fenchel Young Inequality. In the same way the polar of a cone reverses inclusion, convex conjugates reverse majorization, i.e., $f > g$ then $f^* < g^*$. There is also a rich set of calculus rules which can be used to manipulate and derive conjugates. For example, $f(\cdot + a)^* = f^* - (a^T \cdot)$ and $(\lambda f)^* = \lambda f^*(\lambda^{-1} \cdot)$. For a full exploration of these calculus rules, refer to Rockafellar and Wets [RW98, Chapter 11]

2.2.1. infimal convolution. The conjugate of the sum of two functions is of particular interest. Indeed, for closed convex functions f and g, if $\text{ri}(\text{dom}(f)) \cap \text{ri}(\text{dom}(g))$ is nonempty then $(f+g)^* = f^* \otimes g^*$ [Roc70, Theorem 16.4] where $\otimes$ is the *infimal convolution* or *epi-addition*,

$$(f \otimes g)(z) = \inf_{x+y=z}\{f(x) + g(y)\} = \inf_x\{f(x-z) + g(x)\}.$$

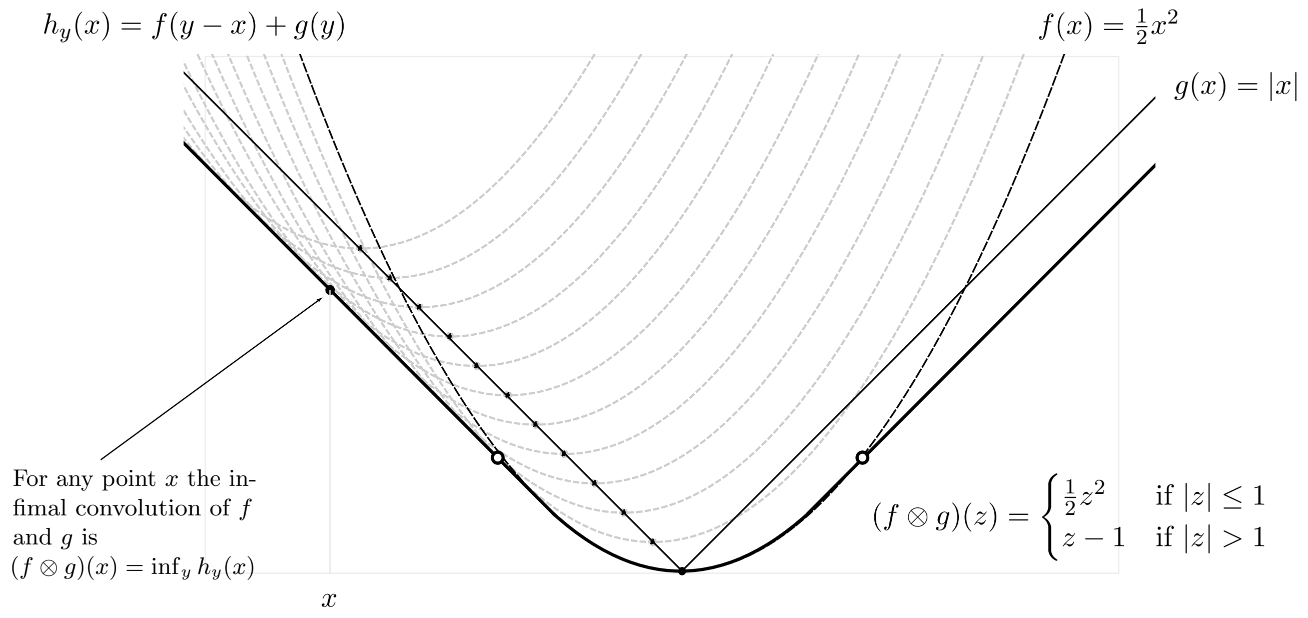


FIGURE 2.2.1. The Moreau envelope of the 1-norm

The name infimal convolution comes from the operator's resemblance to the familiar convolution operator, and epi-addition comes from the observation that $\mathrm{epi}(f \otimes g) = \mathrm{epi}(f) + \mathrm{epi}(g)$. Of particular significance is the infimal convolution of a convex function f with the quadratic function $\frac{1}{2}\|\cdot\|^2$, the *Moreau envelope* $\mathbf{env}_f$ and the *proximal operator* $\mathbf{prox}_f$,

$$\mathbf{env}_f(x) = f \otimes \tfrac{1}{2}\|\cdot\|^2 = \min_z\{\tfrac{1}{2}\|z-x\|^2 + f(z)\}, \tag{2.2.1}$$

$$\mathbf{prox}_f(x) = \operatorname*{argmin}_z\{\tfrac{1}{2}\|z-x\|^2 + f(z)\}. \tag{2.2.2}$$

The proximal operator is of particular interest to us as it is a generalization of the projection—indeed for $\mathbf{env}_{\delta(\cdot|S)}$ is the squared distance to S, and $\mathbf{prox}_{\delta(\cdot|S)}$ the projection onto S. The proximal operator, therefore, plays the same role the projection does in many convex optimization algorithms. We devote Chapter 9 to the study of its computation for scaled $g = \frac{1}{2}\|\cdot\|_H^2$.

2.3. Fenchel duality

Finally, we are ready to approach Fenchel duality, a framework for constructing duals for problems of the form

$$(\mathbf{P}) \qquad \underset{x}{\text{minimize}} \qquad f(x) + g(Ax)$$

We refer to the "original" problem as stated above a problem stated in the above as the primal $(\mathbf{P})$ for which we can construct a dual, another optimization problem which we can use to deduce certain properties of the primal. We motivate this by searching for a lower bound for the optimal value of $(\mathbf{P})$. We begin with the observation that an convex function's biconjugate always proves a lower bound on f, i.e. $f^{**} < f$ and $g^{**} < g$. Therefore, writing the biconjugate in conjugate form, we get the following series of inequalities

$$\begin{aligned}\inf_x \{f(x) + g(Ax)\} &\geq \inf_x \left\{ \sup_u \{x^T u - f^*(u)\} + \sup_v \{(Ax)^T v - g^*(v)\} \right\} \\ &\geq \sup_{u,v} \left\{ \inf_x \{x^T(u + A^T v)\} - f^*(u) - g^*(v) \right\} \\ &= \sup_u \{-f^*(A^T u) - g^*(u)\}.\end{aligned}$$

The second to last inequality comes from the fact that $\inf_x \sup_y f(x,y) \geq \sup_y \inf_x f(x,y)$. This is *weak duality.*

Under certain constraint qualifications, primal-dual pairs of problems exhibit *strong duality*, where the optimal value for both problems match, and the primal and dual programs play symmetric roles. The following theorem gives the conditions for strong duality.

Theorem 2.3.1. *Strong Fenchel Duality [Ber09, Prop. 5.3.8]* Let f and g be convex functions, and A be a $m \times n$ matrix. Then if f^* is finite and $A \cdot \mathrm{ri}(\mathrm{dom}(f)) \cap g \neq \emptyset$

$$\inf_x \{f(x) + g(Ax)\} = -\inf_v \{f^*(-A^T v) + g^*(v)\}, \tag{2.3.1}$$

Furthermore the pair (x, v) is optimal only if $x \in \mathrm{argmin}_x \{f(x) + v^T Ax\}$.

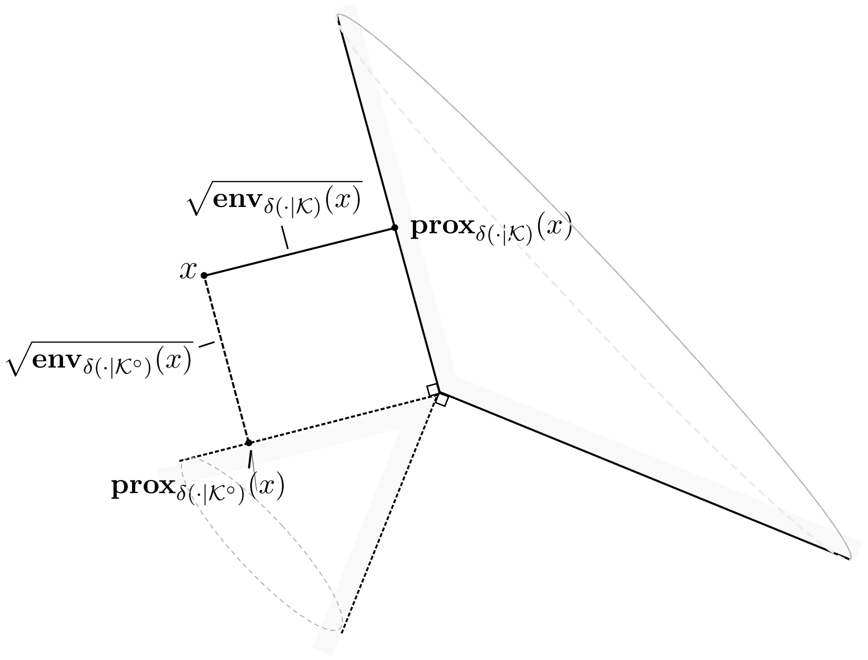

$\sqrt{\mathbf{env}_{\delta(\cdot|\mathcal{K})}(x)}$
$\mathbf{prox}_{\delta(\cdot|\mathcal{K})}(x)$
x
$\sqrt{\mathbf{env}_{\delta(\cdot|\mathcal{K}^\circ)}(x)}$
$\mathbf{prox}_{\delta(\cdot|\mathcal{K}^\circ)}(x)$

FIGURE 2.3.1. Geometric illustration of generalized Moreaus decomposition

Example 2.3.2 (Example: Moreau decomposition)**.** We apply Fenchel duality to the proximal operator, defined in equation (2.2.2). By using the identity $(\frac{1}{2}\|\cdot - b\|^2)^*(v) = \frac{1}{2}\|v-b\|^2 - \frac{1}{2}\|b\|^2$, and assuming that $\mathrm{ri}(\mathrm{dom}(f))$ is not empty, we have strong duality and hence

$$\mathbf{env}_f(x) + \mathbf{env}_{f^*}(x) = \frac{1}{2}\|x\|^2, \qquad \mathbf{prox}_f(x) + \mathbf{prox}_{f^*}(x) = x. \tag{2.3.2}$$

When f is an indicator on a subspace, we recover the Pythagorean equation, equation 2.0.1. When f is an indicator on a convex cone, f^* is the indicator on the polar cone, and hence we have a Pythagorean inequality for cones—this is illustrated in Figure 2.3.1.

2.4. Lagrange duality

The classical calculus method of Lagrange multipliers relates the solution of a constrainted optimization problem for smooth f_i

$$\text{minimize} \quad f_0(x) \qquad \text{s.t} \quad f_i(x) = 0. \tag{2.4.1}$$

to the stationary points of the Lagrangian,

$$L(x,v) = f_0(x) + \sum_{i=1}^{k} v_i f_i(x).$$

Modern convex programming takes a different, game-theoretic interpretation of this classical result. It has its genesis with Kuhn and Tucker [KT51], and starts with the observation that when f_0 is convex and f_i's are linear, for $i \geq 1$, we can think of the stationary points of L as the equilibrium points of a two player game, where (x^*, v^*) satisfy

$$L(x, v^*) \leq L(x^*, v^*) \leq L(x^*, v) \qquad \text{for all } v, x.$$

This interpretation admits a straightforward generalization to convex programs with inequality constraints. For the convex program

$$\text{(2.4.2)} \qquad \underset{x}{\text{minimize}} \quad f_0(x) \qquad \text{s.t} \quad f_i(x) \leq 0,$$

one can similarly characterize all its solutions by looking for the equilibrium points of L on the set $v \geq 0$, with the caveat that the functions f_i satisfy certain *constraint qualifications.* First observe that $\sup_{v \geq 0} L(x,v) = f_0(x) + \delta(x \mid \{\bar{x} \mid f_i(\bar{x}) \leq 0 \; \forall i\})$, therefore the solutions to the game where v plays first are exactly the solutions of problem (2.4.2). By interchanging the sup and inf, we can find a lower bound on the optimal value of problem (2.4.2),

$$\inf_x \{f_0(x) \mid f_i(x) \leq 0\} = \inf_x \sup_{v \geq 0} L(x,v) \geq \sup_{v \geq 0} \inf_x L(x,v),$$

and this allows the construction of the *Lagrange dual*,

$$\underset{v}{\text{minimize}} \quad g(v) := \inf_x L(x,v) \quad \text{s.t.} \quad v \geq 0,$$

which will always prove a lower bound on the original program. This is *weak duality.* However, an equilibrium point requires $\inf_x \sup_{v \geq 0} L(x,v) = \sup_{v \geq 0} \inf_x L(x,v)$, i.e. *strong duality*, which is realized only under certain conditions.

This form of duality, though formally different from Fenchel duality described earlier, turns out to be equivalent, hence saving us the burden of having two expositions on the subject [Mag74]. We show how to derive this from Fenchel duality by noting that if strong duality holds,

$$\begin{aligned}
\inf_x\{f_0(x) \mid f_i(x) \leq 0\} &\overset{(1)}{=} \inf_{x,y}\{f_0(x) + \delta(x|\{x \mid f_i(x) \leq y\}) + \delta_+(y)\} \\
&\overset{(2)}{=} \sup_{x,y,u,v}\{u^T x + v^T y - f_0(x) - \delta(y|\{y \mid f_i(x) \leq y\})\} + 0^*(u)\} \mid v \leq 0, u = 0\} \\
&= \sup_{u,v}\{u^T x - f_0(x) + \sup_{x,y}\{v^T y - \delta(x|\{x \mid f_i(x) \leq y\})\} \mid v \leq 0, u = 0\} \\
&= \sup_v\{\textstyle\sum_{i=1}^k v_i f_i(x) - f_0(x) \mid v \leq 0\}.
\end{aligned}$$

The first equality (1) comes from the fact that $\{f_i(x) \leq y\} \subseteq \{f_i(x) \leq 0\}$, and hence there will always exist an optimal solution to problem (2.4.2) with $y = 0$. Equality (2) comes from an application of Fenchel duality, with the correspondence

$$f(x, y) = f_0(x) + \delta(y|\{x \mid f_i(x) \leq y\}) \qquad \text{and} \qquad g(x, y) = \delta(y \mid \mathbf{R}_+).$$

Thus we can recover the constraint qualifications for strong duality from Fenchel duality, since $\mathrm{dom}(h_0) = \{(x, y) \mid f_0(x) < \infty, f_i(x) \leq y\}$ and $\mathrm{dom}(h_1) = \{(x, y) \mid y \geq 0\}$, therefore the conditions for strong duality are such that

$$\mathrm{ri}(\{(x, y) \mid f_0(x) < \infty, f_i(x) \leq y\}) \cap \{(x, y) \mid y \geq 0\} \neq \emptyset.$$

Part 2

The anatomy of a linear model

CHAPTER 3

Optimization for machine learning

Modeling is the endeavor of translating qualitative phenomena into a quantitative one. Often one begins with a question, real world need, or a goal, stated in a vague and ill-defined manner. The task of modeling is taking this goal, viewing it through the prism of expert opinion, and rendering it into quantitative form—a *model*. A model, for the purposes of this thesis, is a set of plausible explanations (parameterized by some vector x) that have some hope of being predictive of unseen data.

In certain statistical frameworks, such as maximum likelihood [HTF01], the model is chosen based on beliefs about the data generating process. Maximum likelihood posits that the data is generated from a random process for which some true signal x^{true} can be recovered from noisy measurements. Other frameworks, such as computational learning theory, look for solutions that have the capacity to generalize under certain assumptions on the complexity of the data [VV98]. Discussion of these recovery results are beyond the scope of this thesis, but in this chapter, we provide an overview of the optimization problems arising from machine learning and give motivation for the algorithms discussed in later sections.

3.1. Introduction to supervised learning

In this thesis, we are concerned with the *supervised learning problem.* We are given a set feature-label tuples, the *training set* (a_i, b_i), where $a_i \in \mathbf{R}^n$ represents features we believe to be predictive of the labels b_i. Our goal is to learn a predictive function,

$$\mathbf{model}_x(a) : \mathbf{R}^n \to \text{label space}$$

that takes in a (possibly) unseen feature a and gives us its best guess of what the label will be. Our predictive function is parametrized by x, and our goal is to find the best x that represents the data.

We consider two categories of label spaces. In regression, the outcome we wish to measure is quantitative, and the label space is $\mathbf{R}$. In classification, the outcome is categorical, and hence the label space are positive and neative labels, $\{-1, 1\}$, or the interval $[0, 1]$ which stand for the labels in the former, the probability of a having a positive label in the latter.

The foremost goal of any model is that it must fit the data,

$$\mathbf{model}_x(a_i) \approx b_i.$$

The question of how it should fit the data is quantified in the *loss*. The loss, as the name suggests is our expression of how close $\mathbf{model}_x(a_i)$ is from b_i, and examples of this are discussed in Section 3.1.1. If this quantity is minimized, and the loss is small, our model has passed the first test of sanity, but our job is not yet complete. In machine learning, we do not want our model just to do well on the training set, but, in principle, everywhere—this is the problem of generalization.

If we had a large amount of data relative to the size the model, minimizing loss is often sufficient to get a good predictive function. Under certain strict assumptions, minimizing the loss alone can be asymptotically efficient; with the caveat that this holds only as the number of observed points go to infinity. See for example Theorem 10.1.12 of [CB02].

When the data is too small, however, (or if our model is too big) minimizing loss alone isn't sufficient. To take an extreme scenario, consider the example where there are so many parameters the model is under-determined. One must choose then between a potentially infinite number of models which all fit the data perfectly. In such a situation, one might wish to pick "the simplest model", or "the model which is closest to my existing beliefs about the true nature of the parameters". These beliefs about x manifest in the *regularizer*. The regularizer controls a tradeoff between simplicity and fit, and, when properly tuned, often greatly improves the model's capacity for generaliztion. We discuss a variety of regularizers in Section 3.1.2.

The loss function and the regularizer, when combined, are sufficient in most cases to yield a workable model. In Section 3.1.3, we depart from the traditional purview of machine learning and consider the challenges faced when a model is deployed as a decision-making agent. In a deployed model, a model's predictions represent actions, and thus it is not sufficient for the model to simply predict well—it has to also balance several competing goals, and act in accordance with some external specifications. In the context of a system giving out loans, for example, the agent may be constrained regarding the total number of loans that can be issued. Such a system might also want to balance accuracy with fairness, by giving loans out at equal rates in certain subpopulations. These challenges are discussed in Section 3.1.3, where we present a novel framework for tackling such issues we refer to as *dataset constraints.* This work was published in [GCGF16].

3.1.1. Loss functions. In this thesis we consider linear models models of the form

$$\mathbf{model}_x(a) = \mathbf{link}(a^T x).$$

These are models that consist of a linear function composed with **link**, a term we borrow from the language of generalized linear models [McC84]. The linear predictor $a^T x$ expresses our model's beliefs, as it varies in a single dimension, and the link function rescales this belief into the label space—the form the data is presented to us.

Each link function is augmented with a loss function, $\mathbf{loss}(\cdot, \cdot)$, that represents the cost of a datum's deviation between $\mathbf{link}(a^T x)$ and b_i. To find the best model, we minimize the weighted sum of the losses across all the data,

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} w_i \mathbf{loss}(a_i^T x, b_i). \tag{3.1.1}$$

The quantities w_i, the *weights*, are a measure of scale, and we could choose to vary this parameter if we could precisely quantify the accuracy of a label b_i—the weights allow data of higher quality to incur a greater penalty when fitted wrongly. Since the total loss is separable, it is sometimes convenient to write it in vectorized form. Define the quantities (A, b, w) as

$$A = [a_1^T; \ldots; a_n^T], \quad b = (b_1, \ldots, b_n) \quad \text{and} \quad w = (w_1, \ldots, w_n). \tag{3.1.2}$$

We sometimes refer to a *dataset* as the tuple (A, b). Let us consider several examples of loss functions that arise in practice.

3.1.1.1. *Linear Regression.* The familiar example of linear regression is a linear model with the link function set to be the identity. The loss function is a penalty on the "vertical" distance from x_i

to the regression line, and thus the loss function can be written as

$$\mathbf{loss}(\alpha, \beta) = \varphi(\alpha - \beta),$$

and the sum of the losses can be written as a inner product of a separable function operating on the *residual* $Ax - b$.

Squared loss: In the classical theory of maximum likelihood, we assume a data generating process for b_i. Given a_i and our true but unknown model parameters x^{true}, we sample b_i according to the random procedure

$$b_i = a_i^T x^{\text{true}} + \epsilon, \tag{3.1.3}$$

where ϵ is a normally distributed random variable with 0 mean and variance parameter w_i^{-1}. Our goal is to recover x^{true} from noisy pairs b_i, a_i. One way to do this is by maximizing the likelihood function

$$x \mapsto \prod_{i=1}^{m} \exp(-w_i \tfrac{1}{2}(a_i^T x - b_i)^2),$$

since $\exp(-w_i \frac{1}{2}(a_i^T x - b_i)^2)$ is the unnormalized density of the normal distribution. Taking logs on both sides, we obtain the quadratic loss

$$\varphi(\tau) = \tfrac{1}{2}\tau^2, \tag{3.1.4}$$

with the variances playing the role of the reciprocal weights in 3.1.1. The theory of maximum likelihood estimation [CB02, Theorem 10.1.12] gives theoretical guarantees on the consistency of the estimates, i.e. that the optimal solution to the minimization problem approaches x^{true} as the number of independent and identically distributed samples increase. In unweighted vectorized form this loss is the familiar least squares problem,

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|Ax - b\|^2, \tag{3.1.5}$$

and to minimize this loss one need only solve the normal equations $A^T(Ax - b) = 0$.

Robust loss: In 1964 Huber introduced a class of M-estimator [H$^+$64] for situations where the data collected do not satisfy the assumptions of the data generating process, in particular that of (3.1.3). It is common, for example, for the labels b_i to be contaminated by the presence of outliers. The quadratic penalty may pose too strong a penalty for points lying far from the regression line, and the absolute value

$$\varphi(\tau) = |\tau| \tag{3.1.6}$$

may be used instead. These can also be interpreted in the framework of maximum likelihood where the noise in equation (3.1.3) is replaced with a heavier tailed distribution [ABBP11].

Huber Loss: Another M-estimator, the Huber loss [Hub11] combines the robust and squared losses—behaving like a squared loss for small errors, and a robust loss for large errors. The loss is

$$\varphi(\tau) = \begin{cases} \frac{1}{2}\tau^2 & \text{if } |\tau| \leq \epsilon \\ \epsilon(|\tau| - \frac{1}{2}\epsilon) & \text{if } |\tau| > \epsilon \end{cases}. \tag{3.1.7}$$

Note that this loss function is continuous and smooth everywhere, avoiding the non-differential point at zero of the absolute value function. We have encountered the Huber loss as the Moreau envelope of f, see Figure 2.2.1.

Quantile Loss: The quantile loss [Koe05, KBJ78] is useful when there is asymmetry in the cost of a positive and negative error in the regression line. If we would prefer to overestimate or underestimate a certain predictor, we can use the loss

$$\varphi(\tau) = (\epsilon - 1)\max\{0, \tau\} + \epsilon \max\{0, -\tau\}. \tag{3.1.8}$$

Note that when $\epsilon = 0.5$, we recover the robust 1-norm loss.

Vapnick ϵ-Insensitive Loss: The ϵ-insensitive loss, proposed by Vapnick [VV98], is one in which we ignore errors less than ϵ. Errors larger than ϵ can either be penalized linearly

$$\varphi(\tau) = \max\{0, |\tau| - \epsilon\}, \tag{3.1.9}$$

or if a smooth function is preferred, quadratically via the loss

$$\varphi(\tau) = \max\{0, |\tau| - \epsilon\}^2. \tag{3.1.10}$$

This loss may seem contrived but is appealing computationally—models produced with this loss function allow for compact representation in dual form as a linear combination of a small number of representative feature vectors, making it useful in situations where the feature vectors are really a small number of observations lifted to very high dimensions.

It is instructive for us to work through the following simple example of a one-dimensional regression problem.

Example 3.1.1 (Estimates of centrality)**.** Consider the problem of estimating some central value from a population of numbers. This can be thought of as a regression problem where $a_i = 1$ and $b_i \in \mathbf{R}$, where b_i are the values of the population. We will attempt to find a "central" point which minimizes the distance to these values,

$$\underset{x \in \mathbf{R}}{\text{minimize}} \qquad \sum_{i=1}^{m} \varphi(x - b_i).$$

The solution of the above minimization problem, x^*, for the quadratic loss (3.1.4), is the mean, $\frac{1}{m}\sum_{i=1}^{m} b_i$. The optimality conditions for the 1-loss (3.1.6) are that $0 \in \sum \partial|\cdot{-b_i}|(x)$. The minimizers are thus all points x^* which satisfy the condition

$$|\{b_i \mid b_i < x^*\}| = |\{b_i \mid b_i > x^*\}|.$$

If there are an odd number of values, x^* is the median of the values b_i. If there are an even number of values, this includes the median (defined conventionally to be the average of the middle two digits), but also all the values between the central two digits. All optimal points of the quantile loss (3.1.8) satisfy

$$|\{b_i \mid x < x^*\}| = \left(1 - \epsilon^{-1}\right) |\{b_i \mid x > x^*\}|,$$

and thus gives us the ϵ^{th} quantile of a series of numbers.

3.1.1.2. *Binary classification.* The two broad approaches to classification are maximum-margin classification [BGV92, CV95] and maximum likelihood [HL00]. The former is an outgrowth of computational learning theory, while the latter has its origins in frequentist statistics. Though both approaches differ in philosophy, the techniques often result in similar models, and one's preferences for one over the another can sometimes be nothing more than a matter of taste. In general maximum-margin approaches are more scalable—they can be solved very efficiently in their dual representation, and their solutions often admit compact representations. Probabilistic approaches, on the other hand, have advantages in inference. They assume a certain data generating process, and hence one can back out the probabilities of a certain point belonging to a positive or negative class.

Both methods determine classification by the sign of $a_i^T x$ and thus we can write the loss function in the following, abbreviated form

$$\mathbf{loss}(\alpha, \beta) = \varphi(\alpha\beta)$$

since $b_i \in \{-1, 1\}$. For notational simplicity we sometimes use $B = \mathbf{diag}(b)$ and refer to the loss as a separable function of BAx.

3.1.1.3. *Maximum-margin classifiers.* We take an unconventional approach to interpreting maximum-margin classifiers—as upper bounds on the 0-1 loss functions. Maximum-margin methods use the model

$$\mathbf{model}_x(a) = \mathbf{sign}(a^T x),$$

which has a geometrical interpretation of finding a *separating hyperplane* $\{a \mid y^T a = 0\}$ such that all positive labels are on one side of the hyperplane and all negative labels are on the other. In general we wish to minimize the number of misclassified points via the optimization problem

$$\underset{y}{\text{minimize}} \quad \mathbf{1}^T \mathbb{I}(BAx) \qquad \text{where} \qquad \mathbb{I}(\tau) = \begin{cases} 1 & \text{if } \tau > 0 \\ 0 & \text{if } \tau \le 0. \end{cases} \tag{3.1.11}$$

This problem is NP-hard to solve in general [BDEL03, FGRW12], however. Instead, we minimize a convex upper bound on the 0-1 loss. We present several options

Hinge Loss: The hinge loss

$$\varphi(\tau) = \max\{0, 1 + \tau\} \tag{3.1.12}$$

has a simple geometrical interpretation—we penalize the loss according to the distance to the displaced separating hyperplane $\{a \mid y^T a = b_i\}$. One reason we do not choose the more straightforward distance to the separating hyperplane, $\max\{0, \tau\}$, is to avoid the trivial solution at 0.

Quadratic Hinge Loss: A smooth version of the hinge loss [LM01] can be constructed by squaring the hinge loss,

$$\varphi(\tau) = \max\{0, 1 + \tau\}^2. \tag{3.1.13}$$

Though this is convenient computationally due to its smoothness, unlike the hinge loss above, this variation of the hinge is sensitive to the presence of outliers.

Logistic Hinge: The logistic hinge loss [LM01] $\varphi(\tau) = (\log(1 + \exp[-\epsilon\tau])/\epsilon)^2$ is a relaxation of the smooth hinge so it is twice differentiable. The parameter τ is a smoothing parameter, and the loss converges pointwise to the smooth hinge as $\tau \to 0$ [LM01, Lemma 2.1].

These methods earn their namesake when the problem is under-determined, i.e. there are an infinite number of separating hyperplanes that divide the two classes. The choice of the "right" hyperplane was suggested by Vapnick to be the *maximum-margin hyperplane* [VK82], the hyperplane that maximizes the distance from the sum of the closest points in both respective classes.

To find the maximum-margin hyperplane, we first assume that the first element of the feature vector is a distinguished *bias* term, -1. Let $a_i = [-1, \bar{a}_i]$ and let $x = [x_1, \bar{x}]$. We add this feature to allow arbitrary shifts of the separating hyperplane, $\{a \mid \bar{x}^T \bar{a} = x_1, a = [a_1, \bar{a}]\}$. Then, surprisingly, the minimum-norm solution with the bias omitted (it does not make sense to penalize shifts of the separating hyperplane),

$$\underset{x=[x_1,\bar{x}]}{\text{minimize}} \quad \frac{1}{2}\|\bar{x}\|^2 \qquad \text{s.t.} \quad \sum_{i=1}^{n} \varphi(b_i a_i^T x) \le 0. \tag{3.1.14}$$

yields the classifier with the largest margin. A derivation of this result is shown in Figure 3.1.1.

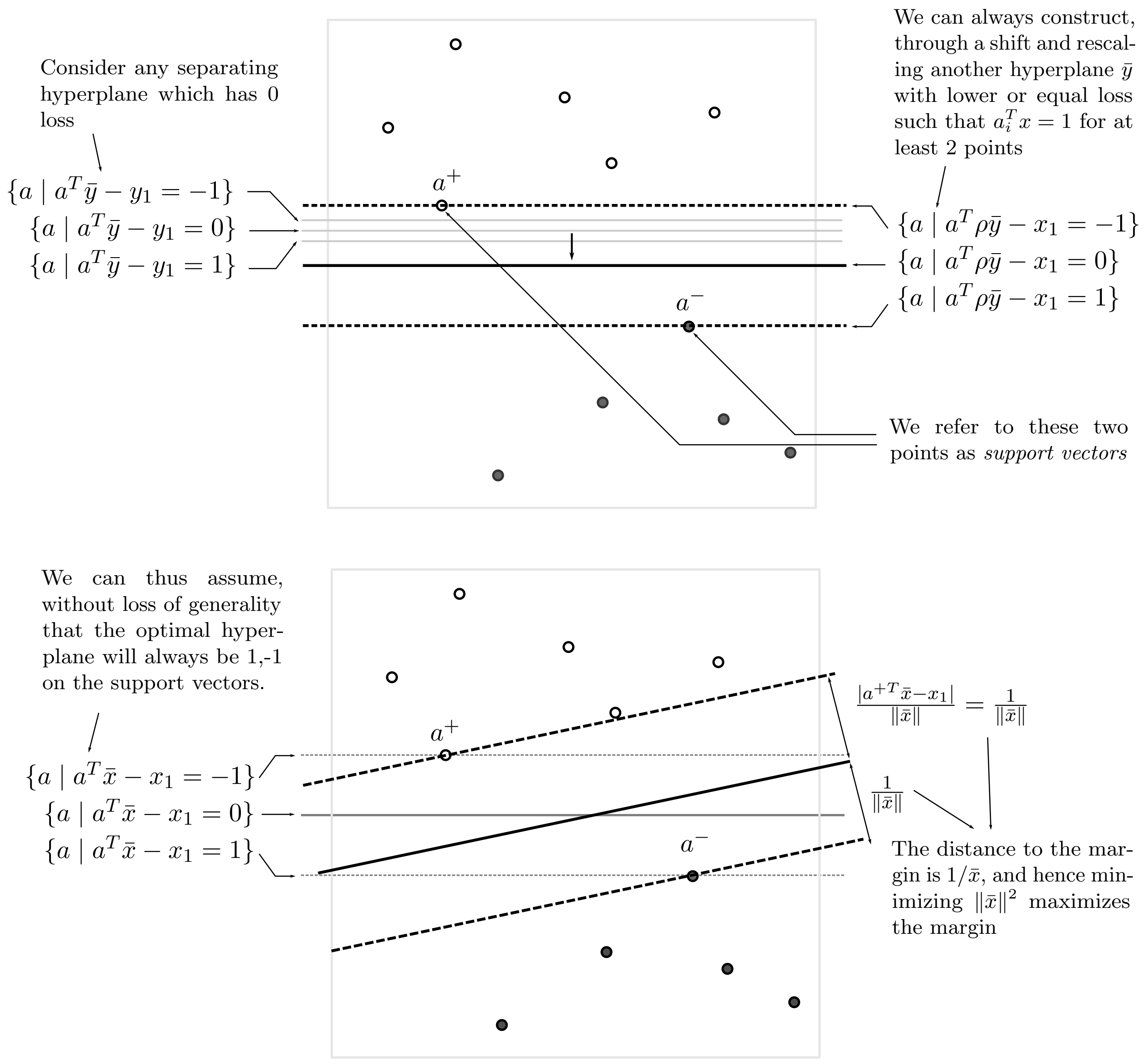

Consider any separating hyperplane which has 0 loss
$\{a \mid a^T\bar{y} - y_1 = -1\}$
$\{a \mid a^T\bar{y} - y_1 = 0\}$
$\{a \mid a^T\bar{y} - y_1 = 1\}$
a^+
a^-
We can always construct, through a shift and rescaling another hyperplane $\bar{y}$ with lower or equal loss such that $a_i^T x = 1$ for at least 2 points
$\{a \mid a^T\rho\bar{y} - x_1 = -1\}$
$\{a \mid a^T\rho\bar{y} - x_1 = 0\}$
$\{a \mid a^T\rho\bar{y} - x_1 = 1\}$
We refer to these two points as *support vectors*
We can thus assume, without loss of generality that the optimal hyperplane will always be 1,-1 on the support vectors.
$\{a \mid a^T\bar{x} - x_1 = -1\}$
$\{a \mid a^T\bar{x} - x_1 = 0\}$
$\{a \mid a^T\bar{x} - x_1 = 1\}$
a^+
a^-
$\frac{|a^{+T}\bar{x} - x_1|}{\|\bar{x}\|} = \frac{1}{\|\bar{x}\|}$
$\frac{1}{\|\bar{x}\|}$
The distance to the margin is $1/\bar{x}$, and hence minimizing $\|\bar{x}\|^2$ maximizes the margin

FIGURE 3.1.1. Derivation of maximum-margin hyperplane from min-norm solutions.

3.1.1.4. *Probabilistic Approaches.* Like linear regression under the squared loss in equation (3.1.3), logistic regression also assumes a probabilistic data generating process for the data. Given a feature vector a and some unknown true parameter x^{true}, we flip independently for every b_i a biased coin [Bis06, Section 4.3.2] with probabilities

$$\Pr(b_i = 1) = \frac{1}{1 - \exp(-a_i^T x^{\text{true}})} \quad \text{and} \quad \Pr(b_i = -1) = \frac{1}{1 - \exp(a_i^T x^{\text{true}})}.$$

Our goal is, from data, to recover the true parameters x^{true}. We do this, following the framework of maximum likelihood, by maximizing the probability of seeing a particular sequence of observations:

$$\underset{x}{\text{maximize}} \qquad \prod_{i=1}^{m} \frac{1}{1 - \exp(b_i a_i^T x)},$$

or equivalent by minimizing its logarithm,

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} \varphi(b_i a_i^T x), \qquad \text{where} \qquad \varphi(\tau) = \log(1 + \exp[-\tau]).$$

This loss function is smooth, twice differentiable, and is typically solved using a smooth solver such as gradient descent or Newton's method. If we had an estimate of x^{true}, we could back out the probabilities of a being in a positive or negative class, and hence

$$\mathbf{model}_x(a) = \frac{1}{1 - \exp(-a^T x)}.$$

An alternative model, probit regression, follows the same framework but determines class probabilities by looking at the sign of a linear regression model,

$$b_i = \mathbf{sign}(a_i^T x + \epsilon_i).$$

Here each ϵ_i is an independently and identically distributed standard normal. The link function for probit regression [Bis06, Section 4.3.5] is equal to the cumulative density of the normal distribution, with a corresponding loss function found by taking its logarithm.

3.1.2. Regularizers. A canonical example of over-fitting arises in polynomial regression. Consider the problem of linear regression, discussed earlier in Section 3.1.1.1 where we wish to fit a sequence of distinct one dimensional feature α_i to real valued outputs, b_i. The lifted feature vector

$$a_i = [1, \alpha_i, \alpha_i^2, \ldots, \alpha_i^k]$$

fits a k-degree polynomial to the data, and when $k = n$, the data can be fit with no error, regardless of α_i and b_i. This perfect fit tells us nothing about the actual relationship between a_i and b_i, whatever they may be, as it is indifferent to the values—we refer to this as *overfitting.* The regularizer

$$\mathbf{reg}(x) : \mathbf{R}^n \to \mathbf{R}_+ \cup \{\infty\},$$

is an additive penalty in the space of the parameters which curbs our model's instincts to overfit. It serves both as a way of encoding expert knowledge about the problem, and as a way of penalizing model complexity. Instead of minimizing the loss function alone, we now minimize

$$\underset{x}{\text{minimize}} \quad \sum w_i \mathbf{loss}(a_i, b_i) + \lambda \cdot \mathbf{reg}(x). \tag{3.1.15}$$

When $\mathbf{reg}(x)$ is real valued, the parameter λ controls the trade-off between model simplicity and fit. When the regularizer is positively homogeneous, the λ is equivalent to a rescaling of the variables, and when the regularizer is just an indicator function the λ serves no function and can be set to 1.

3.1.2.1. *Quadratic Regularizers.* For an underdetermined model, the minimum norm solution—the pseudoinverse in regression, and the maximum-margin hyperplane in classification, (see Section 3.1.14)—is often considered to be the simplest and most parsimonious solution. It is thus sensible to use the quadratic regularizer,

$$\mathbf{reg}(x) = \frac{1}{2}\|x\|^2,$$

as a direct penalty on the magnitude of a solution. The quadratic penalty is typically motivated via a Bayesian interpretation, and it embodies the belief that, in the absence of any knowledge, the model's parameters are normally distributed with mean 0 and variance $1/\lambda$.

Unnaturally large solutions are often a sign of instability in the solution space—and indeed, the problem of overfitting is intrinsically connected to the problem of poor conditioning. Consider the linear regression problem in 3.1.1.1 with squared loss. For the normal equations $A^T(Ax - b) = 0$, we can think of condition number of a matrix A^TA as giving us a way of bounding the magnitude of change in the solution of the system with respect to perturbations in b. To that effect, a poorly conditioned matrix A^TA implies that the solution is highly sensitive to changes in b, the labels. However, if the tiniest change, one possibly induced by a subtle measurement error, causes our model to fluctuate wildly, we have likely begun to fit the noise. Indeed, in the polynomial regression example in the introduction, the Vandermonde matrix $A_{ij} = \alpha_j^i$ (for most values of α_i) is notorious for being poorly conditioned. The quadratic penalty can thus be seen as a way of improving the conditioning of a problem. Combined with linear regression, we now solve the better conditioned normal equations

$$(A^T A + \lambda I)x = A^T b.$$

This is known as Tikhonov regression. Whatever the statistical motivations of adding the quadratic penalty may be, its simplicity and useful conditioning problems make it a sensible default.

3.1.2.2. *Sparse Regularizers.* While having a solution with small norm has its virtues, there are circumstances when a sparse solution is better. If we believe, for example, that certain components of the feature vector have no predictive value, we might wish to eliminate these features altogether. Two highly correlated features, in the same spirit, might give the same information that just one of them does, making the other redundant. The task of pruning redundant features is the problem of *feature selection*, and when done right increases model accuracy, interpretability, and computability.

Classical approaches to this problem include subset selection [HTF01, Section 3.3] and stepwise regression [HL00, Section 4.3]—the former requires a search in the power set of feature vectors, and the latter is a heuristic with dubious statistical properties. Convex optimization furnishes an elegant solution in the form of *sparsity promoting regularizers*, which is both easily computable and has strong theoretical backings (see e.g., [LTTT14] and the references therein).

The 1-norm regularizer, $\mathbf{reg}(x) = \|x\|_1$ is one such example. Because of its nonsmooth structure, it is sparsity promoting, i.e. when λ (3.1.15) is large, solutions of the problem tend to be sparse. When used in regression in concert with the quadratic loss, this is the celebrated LASSO problem [HTF01, Chapter 3.4.2]. And in classification, this appears as either sparse SVMs [BBE$^+$03] or sparse logistic regression [FHT10, YHL12].

Finer control of sparsity can be obtained by using the 2-norm, which promotes group sparsity [YL06, JMBO10]. We organize our variables into possibly overlapping groups and encourage, through the strength of the regularizer, a zeroing out of the entire group simultaneously. This is the group LASSO,

$$\mathbf{reg}(x) = \|x^1\| + \cdots + \|x^p\|, \tag{3.1.16}$$

where x^i is a subset of the variables of x. This is useful when handling complex valued data, say, and one wishes for the real and imaginary parts of the complex number to zero out simultaneously.

More sophisticated schemes of structured sparsity can be designed with overlapping groups, such as hierarchical selection [ZRY09] and more. We refer the reader to the survey of Jenatton, Audibert, and Bach [JAB11] for a thorough treatment of structured sparsity.

If the parameters of our regression correspond to elements of a time series or are pixels of a 2d image [LZOX14], or most generally a graph [CMMP13], we may wish to minimize some measure of smoothness on the graph [CMMP13], i.e. the difference between the values of two adjacent nodes. Let

$$\mathbf{reg}(x) = \|Nx\|_G \quad \text{with} \quad \|z\|_G = \sum_{i=1}^{p} \|z^i\|_2,$$

where z^i is a partition of z and N is an m-by-n matrix. For anisotropic TV and the graph-based 1-norm regularizer, N is the adjacency matrix of a graph, and each partition z^i has a single unique element, so $\mathbf{reg}(x) = \|Nx\|_1$. For isotropic TV, each partition captures adjacent pairs of variables, and N is a finite-difference matrix.

A simple application of the TV regularizer is the *signal denoising problem*, where the labels correspond to dimensions of a signal, and the features a_i are columns of the identity. The signal denoising problem is

$$\underset{x}{\text{minimize}} \qquad \frac{1}{2}\|x - b\|^2 + \lambda\|Lx\|_1, \tag{3.1.17}$$

and the optimal solution tends to be a smooth approximation to b.

3.1.2.3. *Isotonic Regularizers.* Isotonic regularizers offer us a means of enforcing monotonicity in the parameters, i.e. $x_1 \le x_2 \le \cdots \le x_n$. The isotonic regularizer is then the constraint

$$\mathbf{reg}(x) = \delta(Nx|\mathbf{R}_+) = \begin{cases} 0 & \text{if } x_1 \le x_2 \le \cdots \le x_n \\ \infty & \text{otherwise,} \end{cases}$$

where N is the adjacency graph; see previous Section 9.7.2.4. A simple application of this is isotonic regression—given a sequence of data, b_i, we wish to find a monotonically increasing sequence x_i which comes as close to the original sequence as possible. Like in the denoising problem in equation 3.1.17, we use a_i as columns of the identity and solve

$$\underset{x}{\text{minimize}} \qquad \frac{1}{2}\|x-b\|^2 + \delta(Nx|\mathbf{R}_+)$$

for N equal to the adjacency matrix of a path. This can be extended to any number of dimensions by forcing the signal to respect the partial orderings on its respective discretization grid.

3.1.3. Dataset constraints. The objective function of an agent merely trying to understand the world may differ significantly from one trying to act within it. When we treat the output of our predictive function as actions, our objective must now reflect the goals and limitations of the agent making these decisions. We make two observations in this regard.

First, in our experience, a fruitful way of characterizing the difference between an agent and pure learning system is that an agent must juggle an array of conflicting interests, of which minimizing misclassification error is just one. Barring exceptional circumstances, it is impossible to achieve optimality on all objectives at once, and hence a compromise must be reached by navigating the trade off curve between these differing goals. This is the problem of multi-objective optimization.

As a first pass, it is tempting and, in our opinion misguided, to somehow combine all these goals linearly into one super-goal, giving each sub-goal linear weights according to our intuitions. This is the approach of linear scalarization [Mie12, Section 3.1] and is flawed for the apparent reason that these weights will be extremely contentious to set and depend highly on the scaling of these objective functions. Furthermore, it may not be clear beforehand how these weights interact with the final optimized result. A more principled method is to set a primary goal, and then to set tolerances on the secondary objectives that we are willing to endure. This is the approach of ϵ-constraints [Mie12, Section 3.2]. Since these constraints are often measured in meaningful units, this method provides a far more natural way of framing the problem of multi-objective optimization.

The second observation we make concerns the nature of these secondary objectives. The loss function, as we discussed above, is not only a measure of the mismatch but can be interpreted literally as a penalty for making a mistake. We can thus make a subtle redefinition of what a "mistake" might be. Our agent makes mistakes not only when misclassifying a point, but acting in a manner which we find unproductive. It thus stands to reason that we can, similarly, give our model many examples of "right" and "wrong" ways to act, and measure how badly our agent is acting using the same loss functions defined in Section 3.1.1.

We thus arrive at the following formalization. We consider problems defined on several datasets,

$$A^i = [a_1^{iT}; \dots ; a_{m_i}^{iT}], \quad b^i = [b_1^i, \dots , b_{m_i}^i] \quad \text{and} \quad w^i = [w_1^i, \dots , w_{m_i}^i],$$

for which we search for models only within the constraint set

$$\Big\{ x \,\Big|\, \sum_{j=1}^{m_i} w_j^i \mathbf{loss}(a_j^{iT} x, b_i) \le 1 \Big\}. \tag{3.1.18}$$

We call constraints of the form (3.1.18) dataset constraints. We discuss several examples of meaningful constraints we have encountered in practice.

3.1.3.1. *Regression.* Here we narrow our focus to constraints of the form $\left\{ x \mid \|A^i x - b^i\|^2 \le \epsilon_i \right\}$.

Predictive Intervals: Assuming the data generating process of our data follows equation (3.1.4), and let $\chi_{\sigma,m}$ be a percentage point of the chi-square distribution with m degrees of freedom. Then the set

$$\{x \mid \|A^i x - b^i\|^2 \le \chi^2_{\sigma,m}\},$$

is a $1 - \sigma$ confidence interval of parameters x containing x^{true} [SP95, Equation 43]. Thus for any new feature vector a, we can construct upper and lower confidence intervals for our prediction b by maximizing $a^T x$ and $-a^T x$ subject to this constraint. In other words, we can find a predictive interval $[\mathbf{model}^+(a), \mathbf{model}^-(a)]$ where

$$\mathbf{model}^+(a) = \sup\{a^T x \mid \|A^i x - b^i\|^2 \le \chi^2_{\sigma,m}\}, \quad \mathbf{model}^-(a) = \inf\{a^T x \mid \|A^i x - b^i\|^2 \le \chi^2_{\sigma,m}\}$$

For the simple linear regression problem (3.1.5), these formulas can be evaluated in closed form. But this principle can be applied in more sophisticated situations. In isotonic regression, we are given noisy observations (b_i) of a one dimensional monotonic signal

(b^{true}), and we wish to recover the signal. If we instead want monotonic upper and lower confidence bounds [SP95], we can do this by solving

$$b_i^+ = \sup\{x_i \mid \|x-b\|^2 \le \chi^2_{\sigma,m}, x_1 \le \cdots \le x_n\}$$
$$b_i^- = \inf\{x_i \mid \|x-b\|^2 \le \chi^2_{\sigma,m}, x_1 \le \cdots \le x_n\}.$$

where b_i^+ and b_i^+ are the upper and lower envelopes for b^{true}.

Compressed Sensing: Assume the existence of a sparse signal x^{true} probed by noisy measurements

$$b_i = a_i^T x^{\text{true}} + e, \qquad \|e\|_2 \le \epsilon.$$

We wish to recover the signal x^{true} from noisy measurements (a_i, b_i). If x^{true} has exactly k nonzero elements and our measurements a_i are appropriately chosen [CRT06, Don06] then we can recover the true signal with $\lceil k\log(n/k)\rceil$ observations by solving the problem

$$\underset{x}{\text{minimize}} \quad \|x\|_1 \quad \text{s.t.} \quad \|Ax-b\|^2 \le \epsilon.$$

Coverage: If we knew the mean of the unlabeled population A^i with m_i elements was μ_i, we could enforce a constraint on the set

$$\{x \mid \|\mathbf{1}^T A^i x/m_i - \mu_i\|^2 \le \epsilon\}$$

to keep the predictions within ϵ of the mean.

3.1.3.2. *Classification.* In the problem of classification, one can interpret the output of a predictive function to directly control one of two possible decisions—to accept or to deny, to do or not to do. Thus, the most obvious way to constrain our problem is in terms of counts on the number of mistakes made, i.e. we perform optimization on sets of parameters with bounded error measured in terms of the 0-1 loss function. Formally, for label pair (A^i, B^i) this is the set

$$\left\{x \mid \mathbf{1}^T \mathbb{I}(B^i A^i x) \le k\right\}, \qquad \mathbb{I}(x)_i = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i \le 0. \end{cases}$$

We can consider this in greater generality by introducing weighted sums of counts. Such weights arise when certain errors are more costly than others. In medical applications, say, the cost of a false negative is greatly outweighed by the cost of a false positive. Such weighted penalties also arise when our model specifications are laid out most naturally in terms of bounds on the *ratio* of two quantities. For two datasets (A^1, B^1) and (A^2, B^2), a bound on the ratio of counts of both datasets can be turned into a weighted constraint via the transformation

$$\begin{aligned}
\left\{x \,\middle|\, \mathbf{1}^T\mathbb{I}(B^1A^1x)/\mathbf{1}^T\mathbb{I}(B^2A^2x) \le \alpha\right\} &= \left\{x \mid \mathbf{1}^T\mathbb{I}(B^1A^1x) - \alpha\mathbf{1}^T\mathbb{I}(B^2A^2x) \le 0\right\} \\
&= \left\{x \mid \mathbf{1}^T\mathbb{I}(B^1A^1x) - \alpha[m^2 - \mathbf{1}^T\mathbb{I}(B^2A^2x)] \le 0\right\} \\
&= \left\{x \mid \mathbf{1}^T\mathbb{I}(B^1A^1x) + \alpha\mathbf{1}\mathbb{I}(B^2A^2x)] \le \alpha m^2\right\} \\
&= \left\{x \mid [\mathbf{1}, \alpha\mathbf{1}]^T\mathbb{I}(BAx) \le \alpha m^2\right\}
\end{aligned} \tag{3.1.19}$$

where $B = \mathbf{diag}(B^1, B^2), A = [A^1, A^2]$.

The 0-1 loss, however, is merely aspirational. For practical purposes we use an inner approximation to this set instead. We can construct this inner approximation

$$\left\{x \mid w^T\mathbb{I}(BAx) \le k\right\} \supseteq \left\{x \mid w^T\Phi(BAx) \le k\right\}$$

if Φ is defined as the separable map

$$\Phi(x) = [\varphi(x_1); \cdots; \varphi(x_n)], \tag{3.1.20}$$

where φ is an upper bound on the 0-1 loss—see Section 3.1.1.2 for examples. Here are some examples of dataset constraints for classification:

Churn: We define churn as a measure of mismatch between the predictions of two models—A and B. Imagine that model A is 70% accurate, and that model B is 75% accurate. In the best case, only 5% of test samples would be labeled differently, and all differences would be "wins" for classifier B—classifier B classifies every point A did correctly, and then some. In the worst case, model A would be correct and model B incorrect 25% of the time, model B correct and model A incorrect 30% of the time, and both models correct the remaining 45%

of the time. Then 55% of testing examples will be labeled differently. We define the churn rate as the expected proportion of examples on which the prediction of the model being considered (model B above) differs from that of the currently- deployed model (model A). Let

$$A = \text{examples used for comparison}, \qquad B = \text{labels from classifier 1 on } A,$$

and $k = \alpha m$ where m is the height of A. Then the set $\left\{x \mid \sum_i w^T \Phi(BAx) \leq k\right\}$ is the set of parameters where at most k points differ in the two models.

Coverage: In classification, one may wish to control how often a classifier predicts the positive (or negative) class on some distinguished population. For example, one may want to ensure that only 10% of customers are selected to receive a printed catalog due to budget constraints, or perhaps to compensate for a biased training set. If our population is of size m and we desire at most an $\alpha \in [0,1]$ proportion to be classified as positive, the set

$$\left\{x \mid \mathbf{1}^T \Phi(B^i A^i x) \geq \alpha k\right\},$$

where (A^i, B^i) are the feature-label pairs of the distinguished population, ensures that at least α of the population is positive. Coverage was also considered by [MM07], who proposed what they call label regularization, in which one adds a regularizer penalizing the relative entropy between the mean score for each class and the desired distribution, with an additional correction to avoid degeneracies.

Nyanman-Pearson Learning: Requirements of real-world classifiers are often expressed in terms of ratios of the false positives and false negatives, especially when examples are highly imbalanced between positives and negatives. We can handle this problem via Neyman-Pearson classification [SN05, DBS10], in which one seeks to minimize the false negative rate subject to a constraint on the false positive rate. To do this, we let

$$A^1 = \text{positively labeled data}, \qquad A^2 = \text{negatively labeled data}.$$

and solve the optimization problem

$$\text{minimize} \quad \mathbf{1}^T \Phi(A^{1T} x) \qquad \text{s.t.} \quad \mathbf{1}^T \Phi(-A^{2T} x) \leq k.$$

Fairness: A practitioner may be required to guarantee fairness of a learned classifier, in the sense that it makes positive predictions for members of different subgroups at certain rates. For example, one might require that housing loans be given equally to people of different genders. Hardt et al. [HPS$^+$16] identify three types of fairness, specified in terms of ratios of two groups, A^1, B^1 and the general population A^2 and B^2:

$$\left\{ x \;\middle|\; \alpha k - \epsilon \leq \frac{\mathbf{1}^T \mathbb{I}(B^1 A^1 x)}{\mathbf{1}^T \mathbb{I}(B^2 A^2 x)} \leq \alpha k + \epsilon \right\}.$$

These are (i) demographic parity, in which positive predictions are made at the same rate on each subgroup. Here A^1 are the specialized subgroup, A^2 is the entire population and $B^1 = B^2 = \mathbf{diag}(\mathbf{1})$. (ii) equal opportunity, in which only the true positive rates must match. Here A^1 are the positively labeled entries in a subgroup, A^1 are the positively labeled entiries in the population, and $B^1 = B^2 = \mathbf{diag}(\mathbf{1})$, and finally (iii) equalized odds, in which both the true positive rates and false positive rates must match. Here A^1 and B^1 are the labels for the distinguished subgroup, and A^2 and B^2 are the labels for the entire population. This ratio can be converted into two constraints using Equation (3.1.19).

Egregious Examples: For certain classification applications, examples may be discovered that are particularly embarrassing if classified incorrectly. We propose instead simply adding a constraint ensuring that some proportion of a set of such examples is correctly classified, with

$$\left\{ x \mid w^T \Phi(BAx) \leq 0 \right\},$$

where A and b are the set of k examples we must classify correctly.

Finally, we note in passing that if the bound on the set is too strict, we can use a ramp loss instead of a hinge loss. The details can be found in [GCGF16].

Part 3

Cutting plane methods

CHAPTER 4

Epigraphical cutting plane methods

4.1. Introduction

Let $\mathcal{X}$ be a compact convex set with non-empty interior, and let f be a real valued convex function. In this chapter, we discuss cutting plane methods for solving the problem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(x).$$

Cutting-plane methods operate by means of elimination, with successive points strategically placed such that the uncertainty of the location of an optimal solution is maximally reduced. At any iteration, the localization polytope P_k is our expression of this uncertainty—and in traditional cutting plane methods, this is a convex subset of the search space ($\mathbf{R}^n$) that contains at least one optimal point. Each query then generates a new cutting plane that passes through the query point, reducing the volume of P_k by a fixed proportion.

Epigraph cutting plane methods operate on the same principle but quantify both our uncertainty in function value and location of the optimum. The localization polytope thus exists in epigraphical space, $\mathbf{R}^{n+1}$, and its center (t_k, x_k) gives us our new search point x_k. In this new space, each query generates two cuts, a function value cut, which upper bounds the optimum, and a supporting hyperplane, which lower bounds f, see Figure 4.1.1. Of these two cuts, at least one is guaranteed not to contain the query point, i.e., it is *deep*, thus reducing our uncertainty in $\mathbf{R}^{n+1}$, by a fixed proportion.

The epigraph cutting plane method was first proposed by Goffin, Haurie, and Vial [GV99] in the context of the analytic center, and this method appears to have been reinvented in the literature many times, e.g., [BV07, Meh00]. Despite this, a rigorous analysis of the convergence of the epigraph

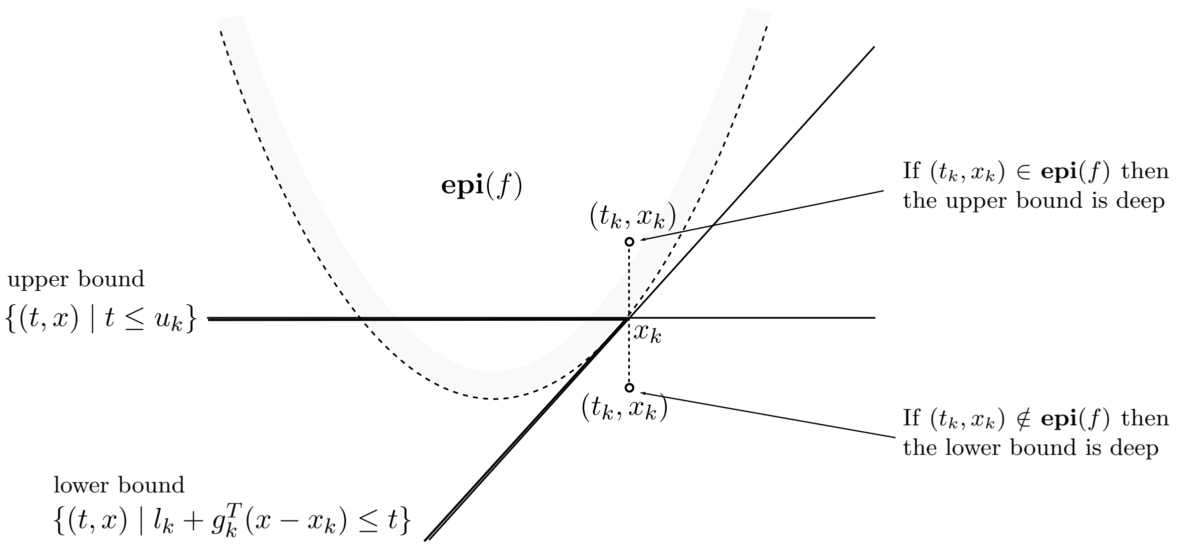

$\mathbf{epi}(f)$
(t_k, x_k)
If $(t_k, x_k) \in \mathbf{epi}(f)$ then the upper bound is deep
upper bound
$\{(t,x) \mid t \le u_k\}$
x_k
(t_k, x_k)
If $(t_k, x_k) \notin \mathbf{epi}(f)$ then the lower bound is deep
lower bound
$\{(t,x) \mid l_k + g_k^T(x - x_k) \le t\}$

FIGURE 4.1.1. Two cuts of the cutting plane method.

cutting plane method still appears to be folklore, and thus we endeavor to describe here a complete analysis.

4.2. Cutting plane methods

We begin with a review of two cutting plane methods: the center of gravity method, outlined in Section 1.2.1.1 and the method of inscribed ellipsoids, outlined in Section 1.2.1.4. The two algorithms are described in a single prototype algorithm, Algorithm 4.1 where the choice of size and **center** determine the which algorithm we use.

Algorithm 4.1: Cutting Plane Method

```
  $\epsilon_{\text{tol}} > 0,\ P_0 = \mathcal{X}$
  for $k \in \{0, 1, \dots\}$ do
1     $x_k \leftarrow \mathbf{center}(P_k)$
2     $g_k \in \partial f(x_k)$
3     $P_{k+1} \leftarrow P_k \cap \{x \mid g_k^T(x - x_k) \le 0\}$
4     if $(\max_{x\in\mathcal{X}} f(x) - \min_{x\in\mathcal{X}} f(x))\,\mathrm{size}(P_k)^{1/n} \le \epsilon_{tol}$ then return $P_k$
  end
```

Center of Gravity Method. The center of gravity method uses the choices

$$\text{size}(P) = \text{vol}(P) := \int_P dz, \qquad \text{and} \qquad \textbf{center}(P) := \text{center of gravity of } P = \frac{\int_P z dz}{\int_P dz}. \tag{4.2.1}$$

This definition is valid when P is bounded and has a nonempty interior, which, assuming $P_0 = \mathcal{X}$, will hold true for any localization polytope.

The center of gravity can be computed in closed form by triangulating the polytope P into simplices, computing the center of gravity for each simplex, and doing a weighted average of these centers. This process, though computationally expensive, is still tractable for problems with dimensions less than 4. The general problem of computing the centroid of a polytope is, however, $\#P$ hard [Rad07] if n is not fixed.

Surprisingly, it is possible to find an estimate of the center of gravity via Monte Carlo. Lovasz and Vempala [LV03] proposed a hit-and-run sampler for finding a uniform point on a convex body with a Markov chain with stationary distribution equal to a uniform random point on the P—and thus the center of gravity can be found by averaging these samples, with its accuracy increasing with the number of samples collected. This method was applied by Bertsimas and Vempala [BV04] in the random walk center of gravity method.

Method of Inscribed Ellipsoids. For a $n \times n$ matrix A and $a \in \mathbf{R}^n$, let $\mathcal{E} = \{Ax + a \mid \|x\|^2 \leq 1\}$ be the maximum inscribed ellipsoid of P. The method of inscribed ellipsoids uses

$$\text{size}(P) = \text{vol}(\mathcal{E}) = \det(A), \qquad \text{and} \qquad \textbf{center}(P) = \text{center of } \mathcal{E} = a. \tag{4.2.2}$$

If our localization polytope is polyhedral, i.e., $P = \{x \mid g_i^T x \leq b_i\}$, the maximum inscribed ellipsoid can be computed in polynomial time by solving the convex program

$$\underset{A \succeq 0, a}{\text{maximize}} \quad \log\det A \qquad \text{s.t.} \quad \|Ag_i\|_2 + g_i^T a \leq b_i. \tag{4.2.3}$$

Since $\log\det(\cdot)$ is smooth on the cone of positive definite matrices, problem (4.2.3) can be solved directly using a proximal gradient method (iteration (1.3.2)), say projected Newton, with a carefully chosen steplength made so as to always remain in the interior of the set of positive definite matrices.

The problem can also be reduced to a linear conic program. First note that the objective $\det(A)^{1/n}$ is convex on the positive semidefinite cone $A \succeq 0$ [AA13, Section 4.3.2]. Thus we use the following characterization of the determinant: for rational numbers $q \in [0, 1/n]$,

$$t \leq \det(X)^q \iff \begin{pmatrix} A & Z \\ Z^T & \mathbf{diag}(Z) \end{pmatrix} \succeq 0 \quad \text{and} \quad (Z_{11}Z_{22}\cdots Z_{mm})^q \geq t.$$

Thus, the optimal solution pair (A, a) in the following convex program

$$\begin{aligned} \underset{t,a,A,Z}{\text{minimize}} \quad & t \\ \text{s.t.} \quad & \|Ag_i\|_2 + g_i^T a \leq b_i \\ & \begin{pmatrix} A & Z \\ Z^T & \mathbf{diag}(Z) \end{pmatrix} \succeq 0 \\ & t \leq (Z_{11}Z_{12}, \ldots, Z_{nn})^{1/n}, \end{aligned}$$

yields the maximum inscribed ellipsoid. The final constraint of a geometric mean can be modeled via an intersection of rotated second-order cones [AA13, Section 3.2.7], thus turning the problem into a linear conic program, amenable to any off the shelf solver that supports inequalities on the semidefinite and second-order cones.

4.2.0.1. *Convergence of both cutting plane methods.* We now provide a theorem that translates the decrease of the volume of P to a concrete bound on the objective value at any point in the iteration.

To offer a unified proof, we first observe the volume of the MIE of P satisfy certain measure-like properties, making it an effective proxy for the volume of P. Indeed, the definition of d-measure, proposed by Tarasov and Khachiyan [Tar88], encapsulates the properties we require in the cutting plane methods of any proxy for volume.

Definition 4.2.1. The function size$(\cdot)$ is a d-measure of a set if it satisfies the following two properties:

(1) $P_1 \subseteq P_2 \Rightarrow \text{size}(P_1) \leq \text{size}(P_2)$ (monotonicity with respect to inclusion)

(2) $\text{size}(a + AP) = \det(A)\text{size}(P)$ (homogeneity with respect to scaling)

Observe, of course, that both the volume of P and the volume of the MIE inscribed in P satisfy the definition of a d-measure. We now proceed with the convergence proof—the following Theroem is an adaptation of the proof of Bubeck [Bub15, Theorem 2.1] generalized to d-measures, described in a remark in a remark in Tarasov and Khachiyan [Tar88].

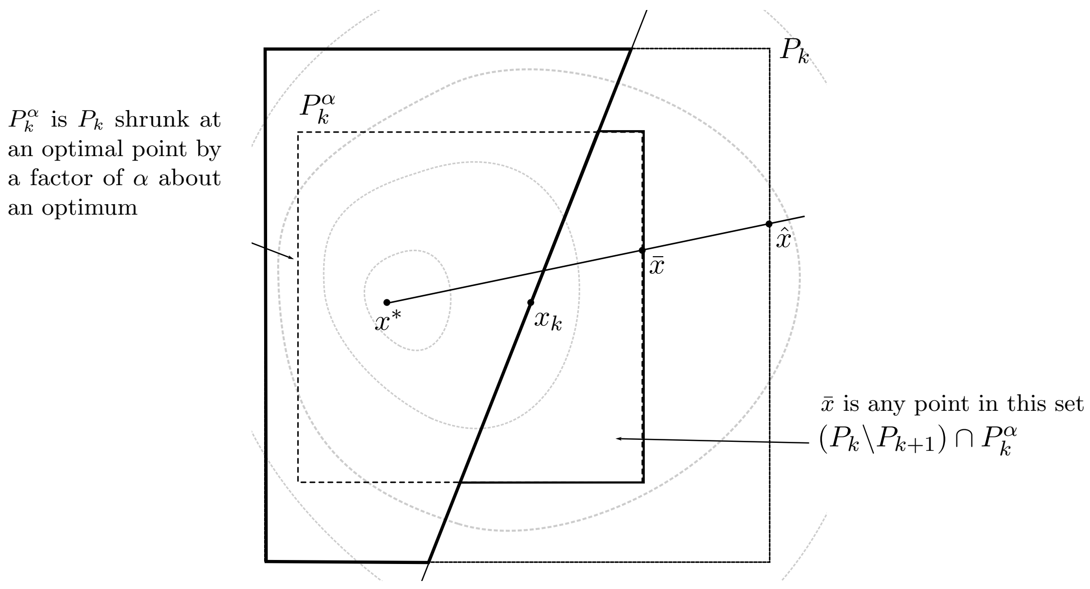

P_k^α is P_k shrunk at an optimal point by a factor of α about an optimum
P_k^α
P_k
$\hat{x}$
$\bar{x}$
x^*
x_k
$\bar{x}$ is any point in this set
$(P_k \backslash P_{k+1}) \cap P_k^\alpha$

FIGURE 4.2.1. Sketch of proof of Theorem 4.2.2.

Theorem 4.2.2. (**Convergence rate for the center of gravity method *and* the method of maximum inscribed ellipsoids**) Let $k \geq 1$ and $x_1, \ldots, x_k$ be the iterates of Algorithm 4.1. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \min_{x \in \mathcal{X}} f(x) \leq \alpha^{\frac{k}{n}} \cdot \left(\max_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} f(x) \right).$$

Here, $\alpha = 1 - \exp(-1)$ for the center of gravity method, and $\alpha = 0.843$ for the method of inscribed ellipsoids.

PROOF. We prove this by induction. Let the total variation of f in P_k be defined by

$$\delta_k = \sup_{x \in P_k} f(x) - \inf_{x \in P_k} f(x),$$

and $P_k^\epsilon = \{x \mid (1-\epsilon)x^* + \epsilon P_k\}$ where x^* is any optimal solution. Then,

$$\mathrm{size}(P_{k+1}) \overset{(1)}{\leq} \alpha \cdot \mathrm{size}(P_k) \overset{(2)}{=} \mathrm{size}(P_k^{\alpha^{1/n}}),$$

where (1) comes from equation (1.2.6) for the center of gravity method and equation (1.2.10) for the method of inscribed ellipsoids, and (2) comes from the homogeneity of size($\cdot$), Definition 4.2.1, and the definition of P_k^ϵ.

Thus, there must exist, by the monotonicity of size($\cdot$) (Definition 4.2.1), a point $\bar{x}$ in $(P_k \backslash P_{k+1}) \cap P_k^\alpha$. Let $\hat{x}$ be point in $P_k \backslash P_{k+1}$. Then the following inquality follows

$$f(x_k) - f(x^*) \overset{(a)}{\leq} f(\bar{x}) - \min_{x \in \mathcal{X}} f(x) \overset{(b)}{\leq} \frac{\|\bar{x} - x\|}{\|\hat{x} - x\|}[f(\hat{x}) - \min_{x \in \mathcal{X}} f(x)] \overset{(c)}{\leq} \alpha^{\frac{1}{n}}[f(\hat{x}) - \min_{x \in \mathcal{X}} f(x)] \leq \alpha^{\frac{1}{n}} \delta_k$$

(a) comes from the fact that all points in $P_k \backslash P_{k+1}$ are larger than $f(x_k)$ by the subgradient inequality. (b) comes from the convex inequality

$$\frac{f(y) - f(x)}{\|y - x\|} \leq \frac{f(z) - f(x)}{\|z - x\|} \qquad y \in [x, z],$$

implying for any $\hat{x}$ on the boundary of P_k, implying

$$f(\bar{x}) - \min_{x \in \mathcal{X}} f(x) \leq \frac{\|\bar{x} - x\|}{\|\hat{x} - x\|}[f(\hat{x}) - \min_{x \in \mathcal{X}} f(x)],$$

and (c) comes from the definition of P_k^ϵ. Applying this result inductively, we obtain the final result. $\square$

Note that above lemma can be applied with minimal modification to oracles which use shallow cuts, such as those discussed in Section 1.2.2.1.

4.3. Epigraph cutting plane methods

The epigraph cutting plane method looks almost identical to Algorithm 4.1, with two conspicuous differences—the algorithm operates in the lifted space $\mathbf{R}^{n+1}$, as we have observed in equation (1.2.14), yields two cuts per iteration. Like the cutting plane method, there are two variations of this method we discuss, the epigraph center of gravity method, and the epigraph method of inscribed ellipses, invoked by the same choices of **center** and size($\cdot$) as in Section 4.2. Let l_0 and u_0 be lower and upper bounds on f on $\mathcal{X}$, respectively:

$$l_0 \leq \min_{x \in \mathcal{X}} f(x) \leq \max_{x \in \mathcal{X}} f(x) \leq u_0, \tag{4.3.1}$$

and

$$\kappa := \frac{u_0 - l_0}{\operatorname{size}(\mathcal{K})^{1/(n+1)}}, \tag{4.3.2}$$

where $\mathcal{K}$ is the inner conic approximation

$$\mathcal{K} = \operatorname{conv}\{\{u_0\} \times \mathcal{X}, (f(x^*), x^*)\}. \tag{4.3.3}$$

The prototype epigraph cutting plane method is shown in Algorithm 4.2.

Algorithm 4.2: Epigraphical Cutting Plane Methods

```
  $\epsilon_{\text{tol}} > 0$, $P_0 = [l_0, u_0] \times \mathcal{X}$ where $l_0, u_0$ are defined in (4.3.1)
  for $k \in \{0, 1, \dots\}$ do
1     $x_k \leftarrow \mathbf{center}(P_k)$
2     $(u_k, g_k) \leftarrow \mathbf{oracle}_f(x_k)$
3     $P_{k+1} \leftarrow P_k \cap \{(t, x) \mid t \leq u_k\} \cap \{(t, x) \mid l_k + g_k^T(x - x_k) \leq t\}$
4     if $\kappa \cdot \operatorname{size}(P_k)^{1/(n+1)} \leq \epsilon_{tol}$ then return $P_k$
  end
```

For the epigraphical variants of the cutting plane algorithm, we have

Lemma 4.3.1. (Relationship between size of localization polytope and optimality) For a convex compact epigraphical polytope, i.e.,

$$P = \{(t, x) \mid t \leq u\} \cap \mathbf{epi}(l(\cdot) + \delta_{\mathcal{X}}),$$

where $\min_{x \in \mathcal{X}} f(x) \leq u \leq u_0$, and $l(\cdot)$ is a convex lower bound on f. Then

$$u - \min_{x \in \mathcal{X}} f(x) \leq \left(u_0 - \min_{x \in \mathcal{X}} f(x)\right) \left(\frac{\operatorname{size}(P)}{\operatorname{size}(\mathcal{K})}\right)^{\frac{1}{n+1}} \leq \kappa \cdot \operatorname{size}(P)^{\frac{1}{n+1}},$$

where $\mathcal{K}$ is defined in (4).

PROOF. Let $\mathcal{K}^u = \mathcal{K} \cap \{(t,x) \mid t \le u\}$. Then

$$\operatorname{size}(P) \overset{(a)}{\ge} \operatorname{size}(\mathcal{K}^u) \overset{(b)}{=} \left(\frac{u - \min_{x\in\mathcal{X}} f(x^*)}{u_0 - \min_{x\in\mathcal{X}} f(x)}\right)^{n+1} \cdot \operatorname{size}(\mathcal{K}).$$

Inequality (a) comes from the fact that

$$P_k = \mathbf{epi}(l(\cdot) + \delta_{\mathcal{X}}) \cap \{(t,x) \mid t \le u\} \supseteq \mathbf{epi}(f(\cdot) + \delta_{\mathcal{X}}) \cap \{(t,x) \mid t \le u\} \supseteq \mathcal{K} \cap \{(t,x) \mid t \le u\} = \mathcal{K}^u,$$

where the last equality follows from the assumption that $u \le u_0$. Next apply the homogeneity of size$(\cdot)$, cf. Definition 4.2.1. Equality (b) comes from the fact that $\mathcal{K}^k$ is just $\mathcal{K}^0$ scaled about the base, and the homogeneity of size$(\cdot)$ in Definition 4.2.1 (b). Taking powers of $1/(n+1)$ on both sides, yields the desired result. $\square$

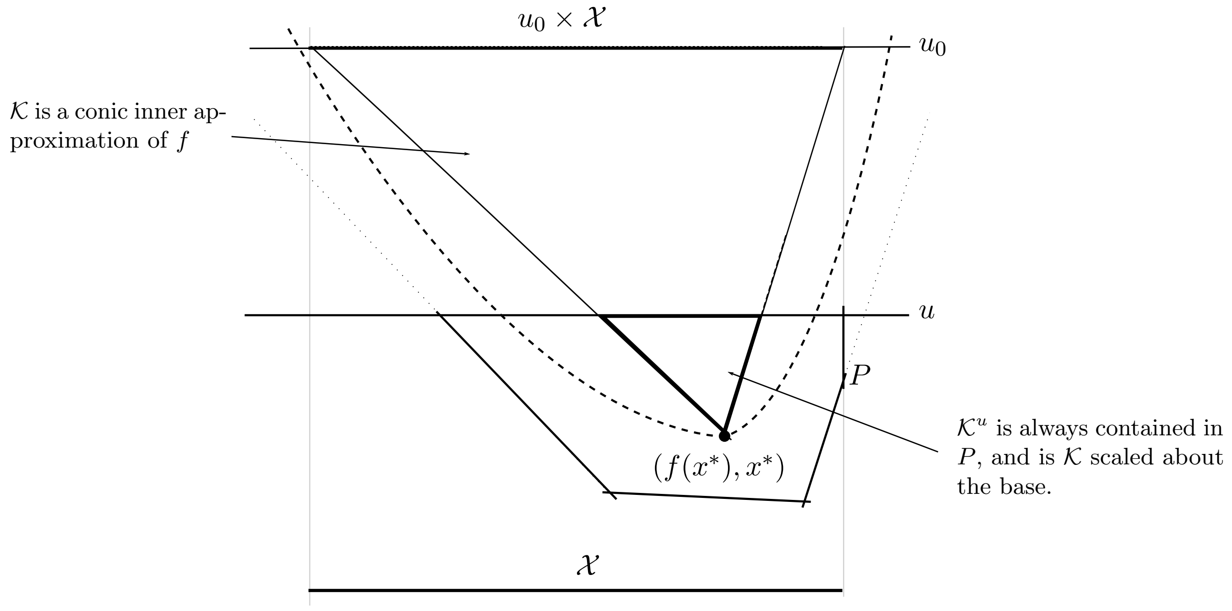

$u_0 \times \mathcal{X}$
u_0
$\mathcal{K}$ is a conic inner approximation of f
u
P
$\mathcal{K}^u$ is always contained in P, and is $\mathcal{K}$ scaled about the base.
$(f(x^*), x^*)$
$\mathcal{X}$

FIGURE 4.3.1. Sketch of proof of Lemma 4.3.1.

First note that P_0 is an epigraphical polytope, with global lower bound $x \mapsto l_0$ and upper bound on the optimum u_0—a property preserved at each iteration. By virtue of the fact that the cuts are deep, the volume reduction guarantees of Grünbaum's theorem, equation (1.2.6) and the theorem of

Tarasov and Khachiyan (1.2.10) [Tar88] guarantees

$$\text{size}(P_k) \leq \alpha^k \text{size}(P_0),$$

for each method's respective α and size$(\cdot)$. Hence by an application of Lemma 4.3.1 we establish the linear convergence of the epigraph cutting plane method.

Theorem 4.3.2. (Convergence of the center of gravity epigraph cutting plane method and epigraph method of maximum ellipsoids) Let $k \geq 1$ and $x_1, \ldots, x_k$ be the iterates of Algorithm 4. Then

$$\min\{f(x_1), \ldots, f(x_k)\} - \min_{x \in \mathcal{X}} f(x) \leq \alpha^{\frac{k}{n+1}} \cdot \left(u_0 - \min_{x \in \mathcal{X}} f(x)\right) \left(\frac{\text{size}(P_0)}{\text{size}(\mathcal{K})}\right)^{\frac{1}{n+1}} \leq \alpha^{\frac{k}{n+1}} \cdot \kappa \cdot \text{size}(P_0)^{\frac{1}{n+1}}$$

Where $\alpha = 1 - \exp(-1)$ for the center of gravity method and $\alpha = 0.843$ for the method of inscribed ellipsoids.

For the center of gravity method, we can bound volumes

$$\left(\frac{\text{vol}(P_0)}{\text{vol}(\mathcal{K})}\right)^{1/(n+1)} \leq \frac{(n+1)^{\frac{1}{n+1}}(u_0 - l_0)}{u_0 - \min_{x \in \mathcal{X}} f(x)} \leq \frac{\sqrt{2}(u_0 - l_0)}{u_0 - \min_{x \in \mathcal{X}} f(x)},$$

with this constant approaching $\sqrt{2}$ as our lower bound on the optimal objective value improves. It is not clear, however, how to do this for general size$(\cdot)$.

Note too that though these bounds are slightly worse than those of the cutting plane methods, they do not account for the fact that we are using two cutting planes for the volume reduction, not one, and that these two planes cuts are typically deep. Since deep cuts cut off more of the polytope than is strictly needed, the volume reduction is significantly greater in practice.

CHAPTER 5

Constructing inexact oracles

In this section we describe several useful examples of *inexact controllable oracles*, **c-oracle**$_f$, (Definition 1.2.7) that arise in practice. Recall a inexact controllable oracle requires three ingredients: given a tuple (x, ϵ), the oracle returns the quantities (l, u, g) where

$$u \qquad \text{is an upper bound on } f(x), \text{ and} \tag{5.0.1}$$

$$l, g \qquad \text{is such that } l + g^T(x - \cdot) \text{ is a global lower minorant of } f \tag{5.0.2}$$

$$\text{where } l - u \leq \epsilon. \tag{5.0.3}$$

For both structures (1.2.1) and (1.2.2), these quantities come from duality. We dualize the objective, f, where

$$f(x) \text{ is either } \quad \max_y h(x, y) \text{ (Section 5.1)} \quad \text{ or } \min_y h(x, y) \text{ (Section 5.2).}$$

The accuracy of the oracle is controlled by refining the primal-dual solutions, and as we discover our tolerance ϵ specifies exactly the accuracy needed in terms of the duality gap.

5.1. Supremal Projection

In this section we focus on f discussed in the introduction (1.2.2), where

$$f(x) := \max_y h(x, y).$$

Access to an optimal solution $y^* \in \operatorname{argmax}_y h(x, y)$ can be used to find a subgradient of f as follows: For any $g \in \partial h(\cdot, y^*)(x)$, we have

$$f(\bar{x}) = h(\bar{x}, y^*) \overset{(a)}{\geq} h(x, y^*) + g^T(\bar{x} - x) = f(x) + g^T(\bar{x} - x) \qquad \text{for any } \bar{x}, \tag{5.1.1}$$

where inequality (a) follows from the subgradient inequality (1.1.1). Thus $g \in \partial h(\cdot, y^*)(x)$ implies that $g \in \partial f(x)$.

5.1.1. Finding the lower bound. Given any point $\bar{y}$, a global affine minorant can be easily constructed via equation (5.1.1). By substituting y^* for $\bar{y}$ in (1.1.1), we obtain

$$f(\bar{x}) = h(\bar{x}, \bar{y}) \geq h(x, \bar{y}) + g^T(\bar{x} - x) \qquad \text{for any } \bar{x}. \tag{5.1.2}$$

Thus, we may take for any $\bar{y}$ a lower bound (5.0.2):

$$l = h(x, \bar{y}) \qquad u \in \partial h(\cdot, \bar{y})(x).$$

5.1.2. Finding the upper bound. An upper bound for $f(x)$, however, is trickier to obtain, and requires assuming some extra structure in $h(x, \cdot)$ and an appeal to duality. Let us, for notational clarity, deemphasize the dependence of h on x by letting $h_x(y) := -h(x, y)$. Furthermore, assume for example that h_x can be written as the sum

$$h_x(y) = \bar{h}(Uy) + \tfrac{1}{2}\|y\|^2. \tag{5.1.3}$$

If we assume that strong Fenchel duality holds, i.e. $U \cdot \mathrm{ri}(\mathrm{dom}(\bar{h}))$ is not empty (see Theorem 2.3.1), we have, for any choice of $\bar{v}$.

$$f(x) = \sup_y h(x, y) = -\inf_y h_x(y) = -\inf_y \{\bar{h}(Uy) + \tfrac{1}{2}\|y\|^2\} \tag{5.1.4}$$

$$\overset{(a)}{=} \inf_v \{\bar{h}^*(v) + \tfrac{1}{2}\|U^T v\|^2\} \leq \bar{h}^*(\bar{v}) + \tfrac{1}{2}\|U^T \bar{v}\|^2. \tag{5.1.5}$$

The equality in (a) comes from strong Fenchel duality 2.3.1. Thus, for a choice of $\bar{v}$ we obtain for (5.0.1)

$$u = \bar{h}^*(\bar{v}) + \tfrac{1}{2}\|U^T \bar{v}\|^2.$$

5.1.3. Controlling the tolerance. In summary, we need to obtain a tuple $(\bar{y}, \bar{v})$ that is feasible for the primal-dual pair:

$$\begin{aligned} &\underset{y}{\text{minimize}} \quad \bar{h}_x(Uy) + \tfrac{1}{2}\|y\|^2 \qquad \text{(primal)} \\ &\underset{v}{\text{minimize}} \quad \bar{h}_x^*(v) + \tfrac{1}{2}\|U^T v\|^2 \qquad \text{(dual)} \end{aligned} \tag{5.1.6}$$

such that their values are within ϵ, our tolerance (5.0.3), of each other. These problems may be solved independently or via a primal-dual solver. We discuss this further in Section 5.3.

Example 5.1.1. (Lagrangian Duals) Given the convex program

$$\underset{y}{\text{minimize}} \quad f_0(y) \qquad \text{s.t.} \quad f_i(y) \le 0, \tag{5.1.7}$$

the dual problem is a minimization problem over the positive orthant,

$$\underset{x \ge 0}{\text{minimize}} \quad f(x) = \max_y h(x,y) = -f_0(y) - x_1 f_1(y) - \cdots - x_n f_n(y),$$

a problem of the form (1.2.2).

Example 5.1.2 (Dataset constraints, Section 3.1.3)**.** A tremendous amount of research over the past decade has been poured into efficient solvers for linear classifiers of the form

$$\underset{x}{\text{minimize}} \qquad \sum w_i \mathbf{loss}(a_i^T x, b_i) + \tfrac{1}{2}\|x\|^2.$$

The product of this concerted effort are highly efficient solvers capable of processing gigabytes of data in minutes on consumer grade hardware. These solvers include Vowpal Wabbit [ACDL14], Liblinear [FCH+08], and SVMLite [Joa99]. Our goal in this example is to show how we can leverage such technology to solve problems which involve dataset constraints, as discussed in Section 3.1.3. We focus in this example on linear classification, but the same ideas apply with minimal modification to regression.

We consider a particular realization using dataset constraints, where the loss function in the constraint matches that of the objective. Given datasets $\{(A^1, B^1), \ldots, (A^k, B^k)\}$, we wish to solve the convex program

$$\underset{y}{\text{minimize}} \quad w^{0T}\Phi(B^0A^0y) + \tfrac{1}{2}\|y\|^2 \quad \text{s.t.} \quad w^{iT}\Phi(B^iA^iy) \le 1 \quad i = 1\ldots n, \tag{5.1.8}$$

as per Section 3.1.1.2, and Φ is defined in equation (3.1.20). We may solve this problem via a dual approach (see Section 2.4). The Lagrangian of problem (5.1.8) is

$$L(x,y) = \mathbf{1}^T x + g_x(y)$$

where g_x takes the form of a weighted support vector machine, with weights now variables of x, which we may interpret as Lagrange multipliers of the constraints in (5.1.8):

$$g_x(y) = w_x^T\Phi(BAy) + \tfrac{1}{2}\|y\|^2 \quad \text{where} \quad w_x^i = \begin{cases} w^0 & \text{if } i = 0 \\ x_i w^i & \text{if } i \ne 0 \end{cases}, \qquad .$$

where $B = [B^0, \ldots, B^m]$ and $A = [A^0, \ldots, A^m]$. Assuming strong duality holds, $\sup_{y\ge 0}\inf_x \mathcal{L}(x,y) = \inf_x \sup_{y\ge 0}\mathcal{L}(x,y)$. Observe that this problem is now of the form (5.1.3), with the correspondence $\bar{h}(z) = w_x^T\Phi(z)$ and $U = BA$. Since $\bar{h}^*(z) = w_x^T\Phi^*(\mathbf{diag}(w_x)^{-1}z)$, we follow the recipes of equations (5.1.2) and (5.1.5) to obtain the upper and lower bounds

$$\begin{aligned} u &:= w_x^T\Phi^*(\mathbf{diag}(w_x)^{-1}v) + \tfrac{1}{2}\|BA^Tv\|^2, \\ l &:= w_x^T\Phi(BAy) + \tfrac{1}{2}\|y\|^2, \quad g = A^TB\mathbf{diag}(w_x)\partial\Phi(BAy) + y, \end{aligned} \tag{5.1.9}$$

where $\partial\Phi$ is shorthand for $\partial\Phi(x) = [g_1, \ldots, g_m]$ where $g_i \in \partial\Phi_i(x)$.

Example 5.1.3. (Dataset Constraints, Support Vector Machine) Let us specialize equations (5.1.9) to the hinge loss. Then,

$$\Phi_i(x) = \max\{0, 1-x\}, \quad \Phi_i^*(x) = \delta(x_i \mid [0,1]), \quad \text{and} \quad \partial\Phi_i = \begin{cases} 0 & \text{if } 1 - x \le 0 \\ -1 & \text{otherwise.} \end{cases}$$

Therefore each evaluation of **c-oracle**$_f$ requires the solution of a weighted SVM, with a dual solver, such that the duality gap dips below a certain tolerance ϵ.

5.2. Parametrized optimization

Consider now the class of structures discussed in (1.2.1), where f is defined implicitly in the optimization problem

$$f(x) := \min_y \; h(x, y),$$

where h is jointly convex in (x, y). If we had access to the optimal solution, a subgradient $g \in \partial f(x)$ can be obtained as follows: assuming all the level sets of h are compact (possibly empty), then by [RW98, Theorem 10.13] we can differentiate under the minimization, and thus $g \in \partial h(\cdot, y^*)(x)$ implies $g \in \partial f(x)$.

5.2.1. Finding the upper bound. The parametrized optimization problem is in some sense the dual of the problem (1.2.2). Computing an upper bound on $f(x)$ is easy: any arbitrary $\bar{y}$ furnishes us with an immediate upper bound, i.e.,

$$f(x) = \inf_y h(x, y) \le h(x, \bar{y}). \tag{5.2.1}$$

5.2.2. Finding the lower bound. It is, however, the lower bound that requires an appeal to duality. As in equation (5.1.3), we treat x as a constant, and define $h_x(y) = h(x, y)$. Now assume that h_x has the following composite structure:

$$h_x(y) = \bar{h}_x(Uy) + \tfrac{1}{2}\|y\|^2. \tag{5.2.2}$$

Then, assuming $\operatorname{int}(U \cdot \operatorname{dom}(\bar{h}_x))$ is nonempty, Fenchel duality gives the following lower bound on f at x:

$$\begin{aligned} f(x) = \inf_y h(x, y) = h_x(y) &= \inf_y \{\bar{h}_x(Uy) + \tfrac{1}{2}\|y\|^2\} \\ &= -\inf_v \{\bar{h}_x^*(v) + \tfrac{1}{2}\|U^T v\|^2\} \ge -\bar{h}_x^*(\bar{v}) - \tfrac{1}{2}\|U^T \bar{v}\|^2 := u \end{aligned}$$

for any choice of $\bar{v}$. As a final step, we linearize $\bar{h}_x^*(\bar{v}) - \frac{1}{2}\|U^T \bar{v}\|^2$ at x to obtain a global affine minorant $l + g^T(x - \cdot)$, where

$$l := -h_x^*(\bar{v}) - \tfrac{1}{2}\|U^T \bar{v}\|^2, \qquad \text{and} \qquad g \in -\partial_x[h_x^*(\bar{v})](x). \tag{5.2.3}$$

We are once again in a situation where we solve a primal-dual pair of optimization problems, identical to problem (5.1.6), with the difference between the upper and lower bound defining the duality gap.

Example 5.2.1 (Bias in a Support Vector Machine)**.** There are two subtle variations of support vector machines. The first is a straightforward generalization of problem (3.1.14), often referred to as the soft margin support vector machine, and has the form

$$\underset{y=[y_1,\bar{y}]\in\mathbf{R}^n}{\text{minimize}} \qquad \mathbf{1}^T\max\{0, \mathbf{1} - B(A\bar{y} - \mathbf{1}y_1)\} + \tfrac{1}{2}\|\bar{y}\|^2, \tag{5.2.4}$$

where A, B is a dataset defined in Section 3.1.1.2. The second SVM variation has the form

$$\underset{y=[y_1,\bar{y}]\in\mathbf{R}^n}{\text{minimize}} \qquad \mathbf{1}^T\max\{0, \mathbf{1} - B(A\bar{y} - \mathbf{1}y_1)\} + \tfrac{1}{2}\|y\|^2. \tag{5.2.5}$$

The absence of the term y_1 in the regularizer of (5.2.4) is critical for ensuring that we can obtain the maximum-margin hyperplane solution in the separable case (see Figure 3.1.1), and also makes the solution invariant to arbitrary displacements in the columns of A. These two properties do not hold when a quadratic penalty is put on the bias term, y_1.

Even though problems (5.2.4) and (5.2.5) differ only by a single variable in the regularization term, this change throws a wrench into large-scale dual solvers (such as LIBLINEAR [FCH+08]) that rely on separability in the dual problem to ensure convergence—removing this term from the primal problem adds a single linear constraint to the dual, causing approaches such as coordinate descent to fail. The sequential minimal optimization [Pla98] algorithm circumvents this limitation by updating two coordinates at a time. This workaround is, however, considerably more time consuming, and has no convergence guarantees.

Our workaround is simple, and is based on formulating (5.2.4) as a parametric optimization problem. With this trick, we solve problem (5.2.4) by solving a sequence of problems of the form (5.2.5). With a slight change of notation, define

$$f(x) = \min_y h(x,y) \qquad \text{where} \qquad h(x,y) = \mathbf{1}^T\max\{0, \mathbf{1} - B(Ay - \mathbf{1}x)\} + \tfrac{1}{2}\|y\|^2,$$

and then observe that (5.2.4) is equivalent to minimizing $f(x)$. The inner problem of minimizing $h(x,y)$ over y is separable, and hence is amenable to any solver that applies to problem (5.2.5).

Note that $h(x,y)$ conforms to the structure in equation (5.2.5), with the correspondence $\bar{h}_x(y) = \mathbf{1}^T \max\{0, (I - xB)\mathbf{1} - y\}$ and $U = A$. We can now apply equation (5.2.1) and equation (5.2.3) to obtain the upper and lower bounds as follows: for any pair (y, v), define

$$\begin{aligned} u &= \mathbf{1}^T \max\{0, (I - xB)\mathbf{1} - y\} + \tfrac{1}{2}\|y\|^2 \\ l &= -[(I - xB)\mathbf{1}]^T v - \tfrac{1}{2}\|A^T v\|^2, \\ g &= \mathbf{1}^T B v. \end{aligned}$$

for $0 \le v \le \mathbf{1}$. This restriction on v ensures that $h^*(v)$ is finite.

5.3. Numerical approaches to obtaining primal-dual pairs

The approaches described in Section 5.1 and 5.2 and are both predicated on having an algorithm that generates a sequence of primal-dual solutions (x_k, v_k) to the pair of primal-dual problems (5.1.6) that converge to their respective solutions. We discuss two generic approaches to constructing such iterates.

5.3.1. Pure Dual Solver. We may first apply a first-order method to the dual problem,

$$\underset{v}{\text{minimize}} \quad \bar{h}_x^*(v) + \tfrac{1}{2}\|U^T v\|^2,$$

and recover a primal solution $y = -A^T v$. This generates a sequence (v_k, y_k) for which the duality gap, the difference between the objective values in the respective problems, approach 0. Thus, the algorithm gives us the upper and lower bounds

$$u_k = \bar{h}_x^*(y_k) + \tfrac{1}{2}\|A^T y_k\|^2 \qquad \text{and} \qquad l_k = -\bar{h}_x(-UU^T y_k) - \tfrac{1}{2}\|U^T y_k\|^2,$$

for which $u_k - l_k \to 0$.

5.3.2. Saddle Point Solver. We can, alternately, attack the saddle point problem directly with a primal-dual first-order method. Observe that the primal problem in (5.1.6) can be written as

$$\inf_y \{\bar{h}_x(Uy) + \tfrac{1}{2}\|y\|^2\} = \inf_y \sup_v \{y^T U v - \bar{h}_x^*(v) + \tfrac{1}{2}\|y\|^2\},$$

and the dual can be written as

$$\inf_v \{\bar{h}_x^*(v) + \tfrac{1}{2}\|U^T v\|^2\} = -\sup_v \inf_y \{y^T U v - \bar{h}_x^*(v) + \tfrac{1}{2}\|y\|^2\}.$$

Because we have assumed strong duality, there exists any primal-dual solution $(\bar{v}, \bar{y})$ of the saddle point system

$$(\bar{v}, \bar{y}) \in \operatorname*{argminmax}_{v,y} \{y^T U v - \bar{h}_x^*(v) + \tfrac{1}{2}\|y\|^2\}.$$

Therefore any algorithm that generates a sequence (v_k, y_k) that converges to any optimal solution of the primal dual problems (5.1.6) respectively provides the requisite sequences of upper and lower bounds. If we use an optimal method, then they converge at a rate of $\mathcal{O}(1/k)$ [CLO14].

CHAPTER 6

The inexact epigraph cutting plane algorithm

In this chapter we present a modification to the epigraph cutting plane method described in Algorithm 4.2, that uses inexact controllable oracles (Definition 1.2.7), described in Section 5. Our generalization, Algorithm 6.1, proceeds in a straightforward way, with the principle modification the use of an inexact controllable oracle in lieu of their exact counterparts.

We discuss two variations of this method—the center of gravity inexact epigraph cutting plane method, and the inexact epigraph method of inscribed ellipsoids. The prototype for both methods is shown in Algorithm 6.1, where the choice of **center**, size, υ and γ are determined by the variant of the cutting plane that is used, and the constant κ is defined in equation (4.3.2).

Algorithm 6.1: Epigraphical Cutting Plane With Error

```
  $P_0 = [l_0, u_0] \times \mathcal{X}$ where $l_0, u_0$ are defined in (4.3.1), $\kappa$ in (4.3.2), $\gamma > 0$.
  for $k \in \{0, 1, \dots\}$ do
1     $(t_k, x_k) \leftarrow \textbf{center}(P_k)$
2     $\epsilon_k \leftarrow \gamma(\min\{u_0, \dots, u_{k-1}\} - t_k)$
3     $(u_k, l_k, g_k) \leftarrow \textbf{c-oracle}_f(x_k, \epsilon_k)$
4     $P_{k+1} \leftarrow P_k \cap \{(t, x) \mid t \leq u_k\} \cap \{(t, x) \mid l_k + g_k^T(x - x_k) \leq t\}$
5     if $\max\{\ \upsilon\epsilon_k\ ,\ \kappa \cdot \text{size}(P_k)^{1/(n+1)}\ \} \geq \epsilon_{tol}$ then return $P_k$
  end
```

Let us draw a few parallels between Algorithm 6.1 and its closest cousin, the inexact level-set bundle method described in Section 1.2.2.2. First, every polytope P_k admits a decomposition as an intersection of two sets, one formed by the intersection of all the lower bounds, the other a halfspace formed by the intersection of the upper bounds. Thus,

$$P_k = \text{epi}(l_k^* + \delta_{\mathcal{X}}) \cap \{(t, x) \mid t \le u_k^*\},$$

where $g_0 = 0$, x_0 is defined as any point in $\mathcal{X}$, and

$$l_k^*(x) = \max_{0 \le i < k} \{l_k + g_i^T(x - x_i)\}, \qquad \text{(lower bound on } f\text{)}$$

$$u_k^* = \min\{u_0, u_1, \dots, u_{k-1}\}, \qquad \text{(upper bound on optimal objective).}$$

The above definitions are almost identical to the lower and upper bounds defined in (1.2.7).

More importantly, our choice of ϵ_k resembles that chosen in the inexact level-bundle method. At each iteration, the inexact level-bundle method sets ϵ_k to be a fraction of the difference $(u_k^* - \min_x l_i^*(x))$, which we refer to as height(P_k):

$$\text{height}(P_k) := \max\{t \mid (t, x) \in P_k\} - \min\{t \mid (t, x) \in P_k\} = u_k^* - \min_{x \in \mathcal{X}} l_i^*(x).$$

This quantity reflects the geometrical view we take in this chapter.

Our choice of γ in Line 2 of Algorithm 6.1, in contrast, is a fraction of the difference $u_k^* - t_k$, which is the distance from **center**(P_k) to the upper bound. Our error rule thus mirrors that of the level-bundle method, despite us arriving at this conclusion through a different route. Like the level-bundle method, our choice of γ controls the trade off between the number of calls to the oracle, against the amount of work demanded of the oracle. The best choice of γ takes into account the trade-off between the cost of an inexact oracle call as a function of ϵ_k and the number of iterations required.

The convergence results of Algorithm 6.1 come in two flavors. The first convergence result, which we call *outer convergence*, shows that we achieve a geometric decrease in d-measure in P_k, and hence achieve linear convergence. This result is the inexact counterpart of Theorem 4.3.2. The second convergence result, which we call *total effort*, takes into account the accuracies we demand of **c-oracle**$_f$. In the analysis of outer convergence, we have assumed implicitly that every call to evaluate a subgradient of f requires a constant amount of work. In this inexact variation of the epigraph cutting plane method, we wish refine the complexity analysis and take into account the total amount of *effort* needed for convergence. The analysis of total effort assumes that a call to **c-oracle**$_f(x_k, \epsilon_k)$ takes $1/\epsilon_k$ amount of effort, and thus, the total effort of the algorithm is the sum of the sequence $1/\epsilon_k$'s over all the iterations required to drive the optimality below the tolerance ϵ_{tol}. Our analysis bounds this sum from above.

6.1. The inexact epigraph center of gravity method

The center of gravity inexact epigraph cutting plane algorithm uses the same choice of operators **center** and size$(\cdot)$ as the center of gravity method, cf. (4.2.1), and the constant

$$\upsilon = \frac{(1-\exp(-1))(n+1)}{\gamma},$$

where γ is any positive number less than 1.

6.1.1. Stopping condition.

Our first result justifies the choice of $\upsilon \cdot e_k$ as a stopping condition in line 5 of Algorithm 6.1. It is clear that the height of the polytope serves as a bound on the optimality of the best solution because

$$\min\{f(x_1), \ldots, f(x_k)\} - \min_{x\in\mathcal{X}} f(x) \le u_k^* - \min_x l_k^*(x) = \text{height}(P_k).$$

Thus, height(P_k) can serve as a measure of optimality. The height of the polytope can be computed directly via the solution of a small linear program, but we avoid this computation by observing that the product $\upsilon\epsilon_k$ is an upper bound for the height of the polytope.

Lemma 6.1.1. (Relating height(P_k) **and** ϵ_k**)** For Algorithm 6.1,

$$\epsilon_k \ge \frac{\gamma \cdot \text{height}(P_k)}{(1-\exp(1))(n+1)} \qquad \text{for all } k \ge 1.$$

Proof. We first construct a conic inner approximation $\mathcal{K}_k$, and a cylindrical outer approximation $\mathcal{C}_k$ of P_k. Let $A_k = \text{lvl}_{l_k^*}(u_k^*) \cap \mathcal{X}$ and $\hat{x}^* \in \text{argmin}_{x\in\mathcal{X}}\, l_k^*(x)$, then

$$\mathcal{K}_k := \text{conv}\{u_k^* \times A_k, (l_k^*(x^*), \hat{x}^*)\} \subseteq P_k \subseteq [l_k^*(x^*), u_k^*] \times A_k =: \mathcal{C}_k.$$

Recall that the center of gravity of P_k is (t_k, x_k). Taking volumes of these sets, we get

$$\operatorname{vol}(P_k) \geq \operatorname{vol}(\mathcal{K}_k) = \frac{\operatorname{height}(P_k)}{n+1} \cdot \operatorname{vol}(A_k),$$

$$\text{and} \qquad \operatorname{vol}(P_k \cap \{(t,x) \mid t \geq t_k\}) \leq \operatorname{vol}(\mathcal{C}_k \cap \{(t,x) \mid t \geq t_k\}) = (u_k^* - t_k) \cdot \operatorname{vol}(A_k)$$

By a mild abuse of notation, $\operatorname{vol}(A_k)$ is overloaded to refer to the volume of the level set in its natural n dimensional ambient space. See Figure 6.1.1 for an illustration of these approximations. The final inequalities come from the volume of a cone, and a cylinder, respectively. Now,

$$\begin{aligned} \frac{\epsilon_k(n+1)}{\gamma \cdot \operatorname{height}(P_k)} &\overset{(a)}{=} \frac{(u_k^* - t_k) \cdot \operatorname{vol}(A_k)}{\operatorname{height}(P_k) \cdot \operatorname{vol}(A_k)/(n+1)} \\ &\overset{(b)}{\geq} \frac{\operatorname{vol}(P_k \cap \{(t,x) \mid t \geq t_k\})}{\operatorname{vol}(P_k)} \overset{(c)}{\geq} 1 - \exp(-1), \end{aligned}$$

where (a) comes from the definition of ϵ_k, and (b) from our inner and outer approximating bounds, in particular the outer bound with $h = t_k$. Inequality (c) follows from Grünbaum's theorem (1.2.6). We obtain the final result by rearranging terms. $\square$

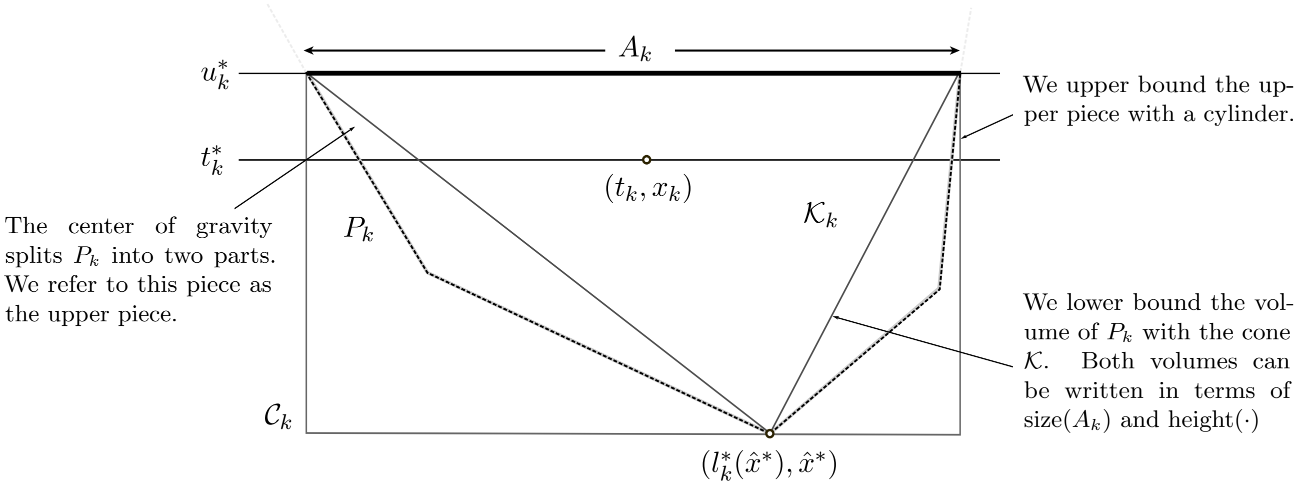

A_k
u_k^*
t_k^*
We upper bound the upper piece with a cylinder.
(t_k, x_k)
$\mathcal{K}_k$
P_k
The center of gravity splits P_k into two parts. We refer to this piece as the upper piece.
We lower bound the volume of P_k with the cone $\mathcal{K}$. Both volumes can be written in terms of size(A_k) and height$(\cdot)$
$\mathcal{C}_k$
$(l_k^*(\hat{x}^*), \hat{x}^*)$

FIGURE 6.1.1. Illustration of outer cylindrical approximation $\mathcal{C}_k$ and inner conic approximation $\mathcal{K}_k$ of $\mathcal{P}_k$

6.1.2. Outer convergence. Next, we move on to the outer convergence of our algorithm. First we describe an important property of the localization polytopes P_k that our method will exploit. Since the level sets of a convex function are nested, and $\{x \mid (t,x) \in P_k\}$ is the level set of l^* lower bound at t, $t_1 \leq t_2$ implies that $\{x \mid (t_1, x) \in P_k\} \subseteq \{x \mid (t_2, x) \in P_k\}$ and thus every vertical half line emanating from inside the polytope will only exit the polytope at the upper bound. We will refer to this as P_k *tapering downwards*. With this definition set, we are now ready to show the guaranteed volume decrease at each iteration.

Lemma 6.1.2. (Convergence of volumes of localization polytopes) For all k,

$$\frac{\mathrm{vol}(P_{k+1})}{\mathrm{vol}(P_k)} \leq 1 - \frac{1-\gamma}{\exp(1)}.$$

Proof. **Case 1: (Deep Cut : $l_k \geq t_k$ or $l_k \leq t_k$ and $u_k \leq t_k$)** The intersection of the cutting planes do not contain (t_k, x_k), therefore by Grünbaum's theorem (1.2.6),

$$\frac{\mathrm{vol}(P_{k+1})}{\mathrm{vol}(P_k)} \leq 1 - \frac{1}{\exp(1)}.$$

Case 2: (Shallow Cut : $l_k < t_k$ and $u_k > t_k$). The shallow cut is illustrated in Figure 6.1.2. By the definition of ϵ, the following inequalities hold:

$$u_k - t_k \leq u_k - l_k \leq \gamma(u_k^* - t_k) \quad \Rightarrow \quad t_k + \gamma(u_k^* - t_k) \geq u_k.$$

Define $\bar{P}_k$ to be the region above the centroid, compressed on the first coordinate, i.e.,

$$\begin{aligned} \bar{P}_k &= \Sigma\big(P_k \cap \{(t,x) \mid t \geq t_k\} - (u_k^*, 0, \ldots, 0)\big) + (u_k^*, 0, \ldots, 0) \\ \Sigma &= \mathrm{diag}(1-\gamma, 1, \ldots, 1). \end{aligned}$$

The bottom-most point of $\bar{P}_k$ is $(t_k + \gamma(u_k^* - t_k), x_k)$, which lies above u_k. Furthermore, because the set tapers downwards, $P_k \cap \{(t,x) \mid t \geq u_k\} \supseteq \bar{P}_k$. Therefore, taking volumes on both sides yields

$$\mathrm{vol}(P_k \cap \{(t,x) \mid t \geq u_k\}) \geq (1-\gamma) \cdot \mathrm{vol}(P_k \cap \{(t,x) \mid t \geq t_k\}). \tag{6.1.1}$$

We therefore obtain the following relationships:

$$\begin{aligned} \mathrm{vol}(P_{k+1}) &\overset{(a)}{\leq} \mathrm{vol}(P_k) - \mathrm{vol}(P_k \cap \{(t,x) \mid t \geq u_k\}) \\ &\overset{(b)}{\leq} \mathrm{vol}(P_k) - (1-\gamma) \cdot \mathrm{vol}(P_k \cap \{(t,x) \mid t \geq t_k\}) \end{aligned}$$

$$\overset{(c)}{\leq} \operatorname{vol}(P_k) - (1-\gamma)e^{-1} \cdot \operatorname{vol}(P_k) = (1 - (1-\gamma)\exp(-1)) \cdot \operatorname{vol}(P_k),$$

where (a) comes from the fact that P_{k+1} is P_k intersected with two cutting planes. The right-hand side of (a) is the volume of P_k intersected with just the upper bound. Inequality (b) comes from (6.1.1), and inequality (c) comes from Grünbaum's theorem (1.2.6). Combining the two results completes the proof. □

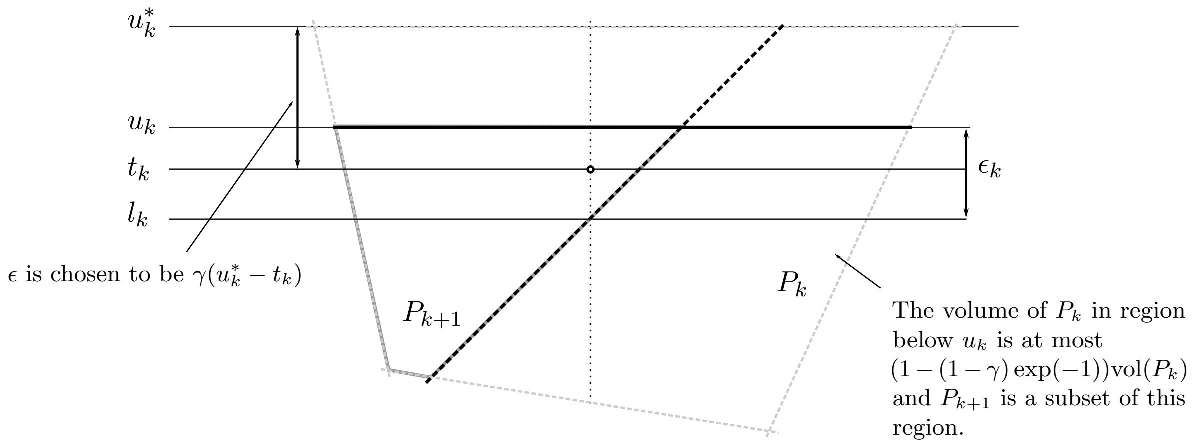

u_k^*
u_k
t_k
l_k
ϵ_k
ϵ is chosen to be $\gamma(u_k^* - t_k)$
P_{k+1}
P_k
The volume of P_k in region below u_k is at most $(1-(1-\gamma)\exp(-1))\operatorname{vol}(P_k)$ and P_{k+1} is a subset of this region.

FIGURE 6.1.2. Illustration of a shallow cut.

Applying the volume decrease inductively, we get

$$\operatorname{vol}(P_k) \leq \left(1 - \frac{1-\gamma}{\exp(1)}\right)^k \operatorname{vol}(P_0),$$

Since Lemma 4.3.1 relates the volumes of P_k to a bound on the objective value, we can invoke it to obtain a outer convergence rate for Algorithm 6.1.

Theorem 6.1.3. (Convergence rate, inexact epigraph center of gravity method) For all k,

$$\min\{f(x_1), \dots, f(x_k)\} - f(x^*) \leq \left(1 - \frac{1-\gamma}{\exp(1)}\right)^{\frac{k}{n+1}} \left(\kappa \cdot \operatorname{size}(P_0)^{\frac{1}{n+1}}\right),$$

where κ is defined in (4.3.2). Therefore, the center of gravity inexact epigraph cutting plane method terminates in no more than

$$\left\lceil \frac{(n+1)\log((\kappa \cdot \mathrm{size}(P_0)^{\frac{1}{n+1}})/\epsilon_{\mathrm{tol}})}{-\log(1-(1-\gamma)/\exp(1))} \right\rceil \in \mathcal{O}\left(n \log \frac{1}{\epsilon_{\mathrm{tol}}}\right)$$

iterations to achieve an optimality of ϵ_{tol} in the objective value.

Note that Theorem 4.3.2 is a special case of Theorem 6.1.3, for the center of gravity method, when $\gamma = 0$.

6.1.3. Total effort. We now focus our attention on the total effort involved in driving a solution to ϵ_{tol}. This is done by noting that the termination condition implies that the work in each iteration will not exceed $(\upsilon\epsilon_k)^{-1}$, and optimizing over γ.

Theorem 6.1.4. (Total effort, inexact epigraph center of gravity method) Assume the algorithm is run till termination. Then the total amount of work involved is

$$\sum \frac{1}{\epsilon_k} \le \frac{n+1}{\gamma\epsilon_{\mathrm{tol}}} \left\lceil (n+1)\log\left(\frac{K}{\epsilon_{\mathrm{tol}}}\right) \right\rceil \in \mathcal{O}\left(\frac{1}{\epsilon_{\mathrm{tol}}} \log \frac{1}{\epsilon_{\mathrm{tol}}}\right).$$

Proof. By the termination condition, and our bound on the total number of iterations in Theorem 6.2.5, the total amount of work is no more than

$$\sum \frac{1}{\epsilon_k} \le \frac{(1-\exp(-1))(n+1)}{\gamma\epsilon_{\mathrm{tol}}} \left\lceil \frac{(n+1)\log(K/\epsilon_{\mathrm{tol}})}{-\log(1-(1-\gamma)/\exp(1))} \right\rceil.$$

Optimizing for γ, we get $\gamma \approx 0.473$. Computing the numerical values and rounding, we obtain the final lower bound. □

6.2. The inexact epigraph method of inscribed ellipsoids

We now proceed to prove analogous results for the center equal to the maximum inscribed ellipse. The the inexact epigraph method of inscribed ellipsoids uses the same choices of **center** and size$(\cdot)$ as the method of inscribed ellipsoids, see equation (4.2.2), with constants $\upsilon = (n+1)/\gamma$ where γ is any positive real number less than 0.081.

6.2.1. Stopping criteria. Our first result is an analog of Lemma 6.1.1, which justifies the stopping condition in Line 5 of Algorithm 6.1.

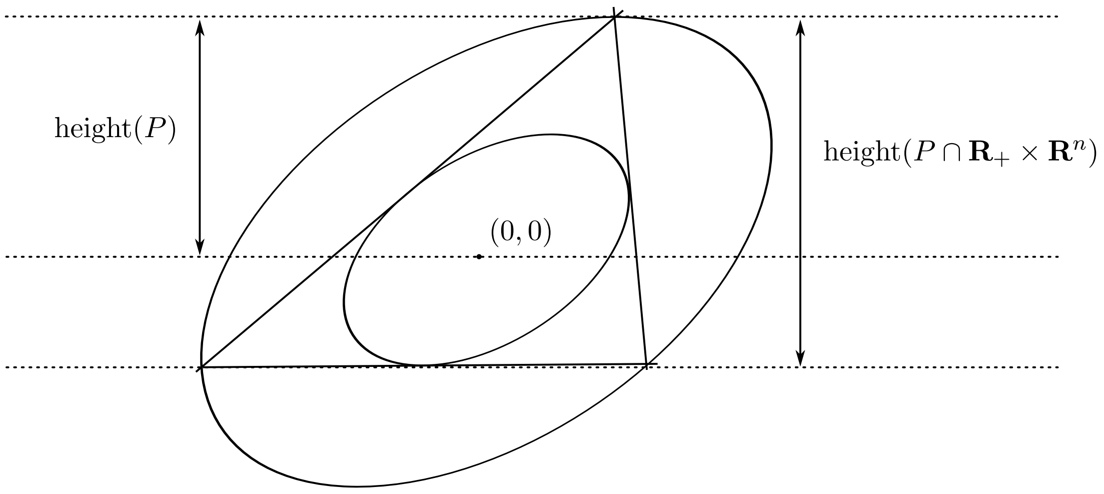


FIGURE 6.2.1. Case where lower bound is tight in $2d$. The $n-$dimensional simplex achieves this bound in general.

Lemma 6.2.1. (Regularity of center$(\cdot)$ **in any dimension)** Let $P \subseteq \mathbf{R}^{n+1}$ be a compact polytope, and assume that the center of the maximum volume ellipsoid of P is a. Then

$$\frac{1}{n+1} \leq \frac{\text{height}(P \cap \{(t,x) \mid t \geq a_1\})}{\text{height}(P)} \leq \frac{n}{n+1}.$$

PROOF. Let $\mathcal{E}$ be the maximum inscribed ellipsoid of P. Assume, without loss of generality that $\mathcal{E}$ is centered at 0. By Remark 2 of Tarasov et al. [Tar88], we know that

$$\mathcal{E} \subseteq P \subseteq n\mathcal{E}.$$

Define the Proj to be the projection of the set P onto the first coordinate, $\text{Proj}(P) = \{t \mid (t, \bar{x}) \in P\}$. Since $\mathcal{E}, P$ and $n\mathcal{E}$ are all convex, their corresponding projections are intervals. Since $\mathcal{E}$ and $n\mathcal{E}$ are both ellipses centered at 0, we can assume then without loss of generality (since the ratios are preserved), that

$$\text{Proj}(\mathcal{E}) := [-1,1], \quad \text{Proj}(P) := [-x_1, x_2], \quad \text{Proj}(n\mathcal{E}) := [-n,n].$$

Using the fact that $\mathrm{Proj}(\mathcal{E}) \subseteq \mathrm{Proj}(P) \subseteq \mathrm{Proj}(n\mathcal{E})$ we have $[-1,1] \subseteq [-x_1, x_2] \subseteq [-n, n]$. Then for $H = \{(t,x) \mid t \geq a_1\})$,

$$\frac{\mathrm{height}(P \cap H)}{\mathrm{height}(P)} = \frac{x_1}{x_2 + x_1} \leq \sup \left\{ \frac{x_1}{x_2 + x_1} \;\middle|\; \begin{array}{c} 1 \leq x_1 \leq n \\ 1 \leq x_2 \leq n \end{array} \right\} = \frac{n}{n+1} .$$

The final equality can be found by solving a simple 2-dimensional optimization problem over a box, for which we can exhaustively search all critical points. Replacing the sup with an inf, we obtain the other bound. □

Lemma 6.2.2. (Relating $\mathrm{height}(P_k)$ **and** ϵ_k**)** For all k,

$$\epsilon_k \geq \frac{\gamma \,\mathrm{height}(P_k)}{n+1}.$$

Proof. Since $u_k^* - t_k = \mathrm{height}(P_k \cap \{(t,x) \mid t \geq t_k\})$, we have from Lemma 6.2.1 and the definition of ϵ_k

$$\epsilon_k = \gamma \,(u_k^* - t_k) \geq \frac{\gamma \,\mathrm{height}(P_k)}{n+1}.$$

This completes the proof. □

6.2.2. Outer convergence. Next we show outer convergence. Here we require a slightly more general variation of result, Theorem γ (sic) of Tarasov et al. [Tar88].

Lemma 6.2.3. Let $\mathcal{E}$ be an ellipse inscribed in P with relative error γ, i.e., $\mathrm{vol}(\mathcal{E}) = (1-\gamma)\mathrm{size}(P)$, where a is the center of this ellipse. Then

$$\frac{\mathrm{size}(P \cap \{x \mid g^T(x-a) \leq 0)}{\mathrm{size}(P)} \leq \frac{0.843}{(1-\gamma)^2}.$$

Armed with this lemma, we are ready to prove the outer convergence of our algorithm. This result is analogous result to Theorem 6.1.2, except the choice of center is here the maximum inscribed ellipsoid of P.

Lemma 6.2.4. (Convergence of volumes of localization polytopes) The volumes decrease at the rate

$$\frac{\text{size}(P_{k+1})}{\text{size}(P_k)} \leq \frac{0.843}{(1-\gamma)^2}.$$

PROOF. We will split the analysis into two cases.

Case 1: (Deep Cut : $l_k \geq t_k$ or $l_k \leq t_k$ and $u_k \leq t_k$.) The intersection of the cutting planes do not contain (t_k, x_k), therefore by Lemma 6.2.3 with $\gamma = 0$,

$$\frac{\text{size}(P_{k+1})}{\text{size}(P_k)} \leq 0.843.$$

Case 2: (Shallow Cut : $l_k < t_k$ and $u_k > t_k$.) In this event, our choice of duality gap ensures that

$$u_k - t_k \leq u_k - l_k \leq \gamma(u_k^* - t_k) \quad \Rightarrow \quad t_k + \gamma(u_k^* - t_k) \geq u_k. \tag{6.2.1}$$

Consider the ellipse formed by the following compressive transformation

$$\bar{\mathcal{E}} = \Lambda\left(\mathcal{E} - (u_k^*, 0, \ldots, 0)\right) + (u_k^*, 0, \ldots, 0)$$
$$\Lambda = \text{diag}([1-\gamma, 1, \ldots, 1]).$$

The center of this compressed ellipse is $(t_k + \gamma(u_k^* - t_k), x_k)$ which lies at or above (u_k, x_k), by (6.2.1). Therefore the horizontal upper bound cuts off the center of $\bar{\mathcal{E}}$. Also observe that $\bar{\mathcal{E}} \subseteq P$ as P tapers downwards, and the vertical region that lies above $\mathcal{E}$ is all contained in P. Furthermore,

$$\text{vol}(\bar{\mathcal{E}}) = \det(\Lambda) \cdot \text{vol}(\mathcal{E}) = (1-\gamma) \cdot \text{vol}(\mathcal{E}).$$

Therefore, $\bar{\mathcal{E}}$ is a $(1-\gamma)$-approximate ellipsoid, and by Lemma 6.2.3

$$\frac{\text{size}(P_{k+1})}{\text{size}(P_k)} \leq \frac{0.843}{(1-\gamma)^2}.$$

Combining these two cases yields the final result. □

Applying this result inductively, we get

$$\operatorname{size}(P_k) \leq \left(\frac{0.843}{(1-\gamma)^2}\right)^k \operatorname{size}(P_0),$$

and thus, combining the above two results, we get a statement on the outer convergence rate of our algorithm.

Theorem 6.2.5. (Convergence rate, inexact epigraph method of inscribed ellipsoids) For all k in Algorithm 6.1,

$$\min\{f(x_1), \ldots, f(x_k)\} - f(x^*) \leq \left(\frac{0.843}{(1-\gamma)^2}\right)^{\frac{k}{n+1}} \left(\kappa \cdot \operatorname{size}(P_0)^{\frac{1}{n+1}}\right)$$

and therefore it terminates in no more than

$$\left\lceil \frac{(n+1)\log((\kappa \cdot \operatorname{size}(P_0)^{\frac{1}{n+1}})/\epsilon_{\text{tol}})}{-\log(0.843/(1-\gamma)^2)} \right\rceil \in \mathcal{O}\left(n \log \frac{1}{\epsilon_{\text{tol}}}\right)$$

iterations to achieve an optimality of ϵ_{tol} in the objective value.

6.2.3. Total effort. Again, we wish to quantify the amount of work involved at each iteration. This theorem is an analog of Theorem 6.1.4, which is identical to it apart from constants.

Theorem 6.2.6. (Total effort, inexact epigraph method of inscribed ellipsoids) Assume the algorithm is run till termination. Then the total amount of work involved, for $\gamma = 0.04$

$$\sum \frac{1}{\epsilon_k} \leq 30 \cdot \frac{n+1}{\gamma \epsilon_{\text{tol}}} \left\lceil (n+1) \log \left(\frac{\kappa \cdot \operatorname{size}(P_0)^{\frac{1}{n+1}})}{\epsilon_{\text{tol}}} \right) \right\rceil \in \mathcal{O}\left(\frac{1}{\epsilon_{\text{tol}}} \log \frac{1}{\epsilon_{\text{tol}}}\right).$$

Proof. From the bound on the total number of iterations, we get

$$\sum \frac{1}{\epsilon_k} \leq \frac{n+1}{\gamma \epsilon_{\text{tol}}} \left\lceil \frac{(n+1)\log((\kappa \cdot \operatorname{size}(P_0)^{\frac{1}{n+1}})/\epsilon_{\text{tol}})}{-\log(0.843/(1-\gamma)^2)} \right\rceil \in \mathcal{O}\left(n \log \frac{1}{\epsilon_{\text{tol}}}\right).$$

Optimizing for γ numerically, we find that the optimum occurs at $\gamma = 0.04$. Substituting this into the equation, we get the final result. □

CHAPTER 7

The parabolic inexact epigraph cutting plane algorithm

7.1. Introduction

In both Chapters 4 and 6, the information we get from the cutting planes are identical. We prove a linear, lower bound on the optimum, restricting the optimum (or the optimal value, optimum tuple) to lie in a certain halfspace. There is no intrinsic reason, however, for us to restrict ourselves to linear lower bounds—in fact, one should always use the strongest lower bound available as this yields the greatest decrease in the volume of the localization set.

When f is strongly convex, and the modulus of strong convexity is known to be μ, we can do better than a linear lower bound. Strong convexty guarantees that every query to $\textbf{c-oracle}_f$ gives a *parabolic* cutting plane, as shown here:

$$\begin{aligned} f(\bar{x}) &\geq f(x) + r^T(\bar{x} - x) - \frac{\mu}{2}\|\bar{x} - x\|^2, && \text{for } r \in \partial f(x) \\ &\geq l + g^T(\bar{x} - x) - \frac{\mu}{2}\|\bar{x} - x\|^2 && \text{for } (u, l, g) = \textbf{c-oracle}_f(x, \epsilon). \end{aligned}$$

While linear lower bounds localize the optimal solution to lie in a particular halfspace, parabolic lower bounds localize the optimal solution to lie in a particular *ball.* The compactness of the region of localization opens many new doors in the design of cutting plane algorithms.

To offer a flavor as to the improvements of a parabolic cutting plane entails, we describe the geometric descent method proposed by Bubeck, Lee, and Singh [BLS15]. Geometric descent stores only two lower bounds at every iteration: The first lower bound is the localization set P_k, which contains the optimum and acts as an outer approximation of all the information obtained by the cutting planes thus far. The next iteration intersects the localization polytope P_k with a second parabolic cutting plane, and the information in this intersection is summarized in the next localization set P_{k+1} via optimal quadratic averaging [DFR16]. Despite the massive amounts of information discarded at every iteration, this method still achieves the optimal convergence rate in the dimension free setting.

Unlike geometric descent, our cutting plane method discards no information at any iteration—we are less concerned with the computational work needed to make a single step as we are with the number of calls to our controllable oracle. Thus we follow the template of Algorithm 6.1 but simply replace our linear lower bounds with parabolic ones. The result is Algorithm 7.1, which differs from Algorithm 6.1 in lines 3 and 5, which instead use a parabolic oracle and a different stopping criterion.

Algorithm 7.1: Epigraphical Cutting Plane With Error

```
  $P_0 = [l_0, u_0] \times \mathcal{X}$ where $l_0, u_0$ are defined in (4.3.1)
  for $k \in \{0, 1, \dots\}$ do
1     $(t_k, x_k) \leftarrow \textbf{center}(P_k)$
2     $\epsilon_k \leftarrow \gamma(\min\{u_0, \dots, u_{k-1}\} - t_k)$
3     $(u_k, l_k, g_k) \leftarrow \textbf{c-oracle}_f(x_k, \epsilon_k)$
4     $P_{k+1} \leftarrow P_k \cap \{(t,x) \mid t \le u_k\} \cap \{(t,x) \mid l_k + g_k^T(x - x_k) + \frac{\mu}{2}\|x - x_k\|^2 \le t\}$
5     if $1000 \cdot n[L^n \text{size}(P_k)^2]^{1/(2+n)} \ge \epsilon_{tol}$ then return $P_k$
  end
```

We require two additional assumptions on f. First, it must admit a quadratic upper bound at the optimum, i.e. for some parameter $L > 0$

$$f(x) \le \frac{L}{2}\|x - x^*\|^2 + \min_{x \in \mathcal{X}} f(x) \qquad \text{for all } x. \tag{7.1.1}$$

Second, for that same L,

$$\mathcal{X} \subseteq \{x \mid L \cdot \tfrac{1}{2}\|x - x^*\| \le u_0\}. \tag{7.1.2}$$

An upper bound of the form (7.1.1) exists if f, for example, is smooth and has L-Lipschitz gradients, e.g., if f is a dual of a strongly convex primal problem, and the second assumption states that the solution to the problem should not lie on the boundary of $\mathcal{X}$.

In this chapter we show these two modifications will enable us to prove a stronger result on the total work involved. Indeed, in contrast to the $\mathcal{O}(\epsilon_{\text{tol}}^{-1} \log \epsilon_{\text{tol}}^{-1})$ amount of work required before (see Theorem 6.1.4), we now require $\mathcal{O}(\epsilon_{\text{tol}}^{-1})$ work. In the interest of brevity, we prove the results in this section for the choices corresponding to (4.2.2), the method of inscribed ellipsoids. However, all the results here generalize to the center of gravity case with minimal modification.

Before we proceed to prove our result, let us consider why a strongly convex function f of the form (1.2.2) might occur in practice.

Example 7.1.1. (Quadratic penalties) We consider functions f of the form $f(x) = \max_y h(x, y)$, where h is convex in x (cf. Section 5.1). We consider the case where h arises from the Lagrangian of the constrained optimization problem (5.1.7), so that

$$h(x, y) = -f_0(y) - x_1 f_1(y) - \cdots - x_n f_n(y).$$

Thus, $f(x) = \max_y h(x, y)$ is the dual objective. Consider what happens if we instead solved a perturbed, strongly convex version of the dual problem

$$\underset{x \geq 0}{\text{minimize}} \quad f^\mu(x) \qquad \text{where} \qquad f^\mu(x) = f(x) + \frac{\mu}{2}\|x\|^2.$$

This corresponds to finding saddle points of the perturbed Lagrangian,

$$h^\mu(x, y) = -f_0(y) - x_1 f_1(y) - \cdots - x_n f_n(y) + \frac{\mu}{2}\|x\|^2.$$

Assuming strong duality, the corresponding primal problem to the dual is

$$-\inf_{x \geq 0} h^\mu(x, y) = f_0(y) + \frac{1}{n\mu}\big(\max\{0, f_1(y)^2\} + \cdots + \max\{0, f_n(y)^2\}\big).$$

And thus we arrive at a dual interpretation for the regularization that adding a strongly convex quadratic in the dual corresponds to replacing constraints with quadratic penalties in the primal. As expected, the stronger the quadratic penality, the lower the coefficient of strong convexity will be. Note that if we have upper bounds on the Lagrange multipliers (our variables x), $0 \leq x \leq \alpha$, then the quadratic penalties turn into Huber penalties. Finally, Assumptions (7.1.1) and (7.1.2) will hold if f_0 is strongly convex, and none of the constraints are redundant.

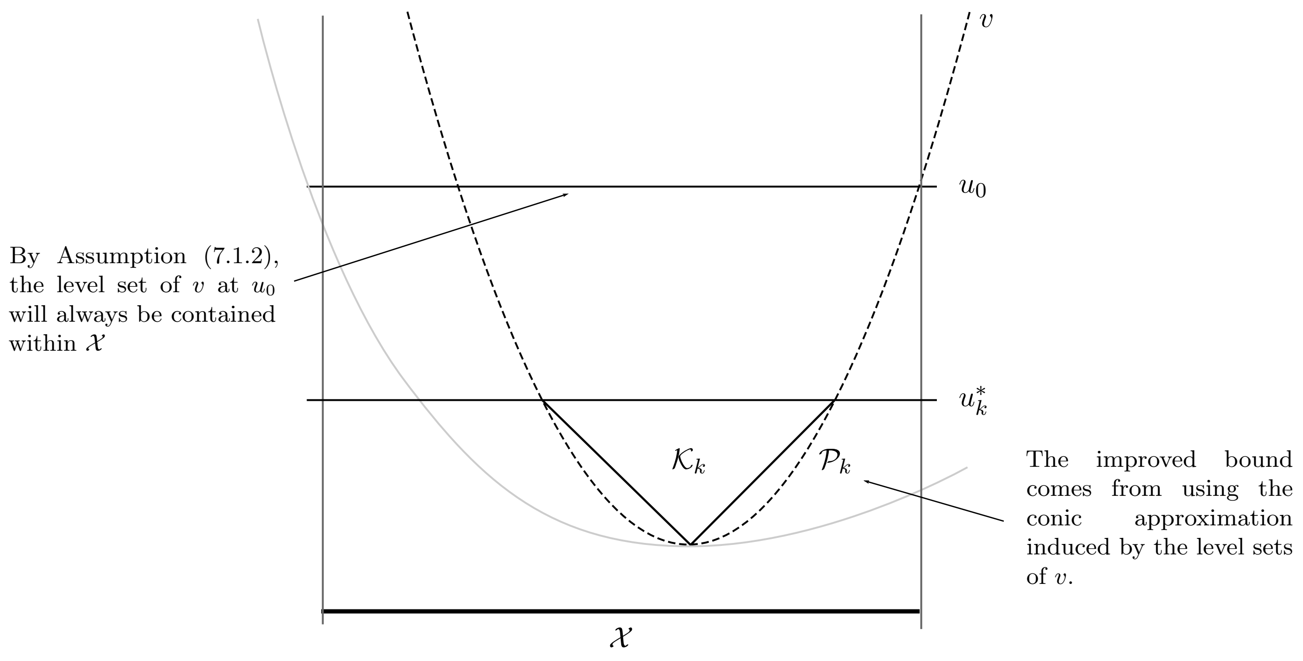

v
u_0
By Assumption (7.1.2), the level set of v at u_0 will always be contained within $\mathcal{X}$
u_k^*
$\mathcal{K}_k$
$\mathcal{P}_k$
The improved bound comes from using the conic approximation induced by the level sets of v.
$\mathcal{X}$

FIGURE 7.1.1. Illustration of inner approximating cones, and the rescaling by Σ_k necessary to transform it into the unit cone.

7.2. The parabolic inexact epigraph method of inscribed ellipsoids

7.2.1. Outer convergence. The assumption that f is upper bounded by a quadratic at the optimum (7.1.1) leads to a quadratic improvement in optimality as a function of the size of the polytope P_k at any iteration.

Lemma 7.2.1. (Relation between optimality and size(P_k)**)** For all k,

$$u_k^* - \min_{x\in\mathcal{X}} f(x) \leq \mathcal{O}(1) \cdot n[L^n \text{size}(P_k)^2]^{1/(2+n)}.$$

where $\mathcal{O}(1)$ is a universal constant.

PROOF. From Assumption 7.1.1 we have the following global quadratic upper bound on f,

$$v := \tfrac{L}{2}\|\cdot - x^*\|^2 + \min_{x\in\mathcal{X}} f(x),$$

i.e., $v \geq f$. Define the two sets $\mathcal{K}_k$, $\mathcal{P}_k$ for which $\mathcal{K}_k \subseteq \mathcal{P}_k \subseteq P_k$,

$$\mathcal{K}_k = \operatorname{conv}\{u_k^* \times \operatorname{lvl}_v(u_k^*), (\min_{x\in\mathcal{X}} f(x), x^*)\}. \quad \mathcal{P}_k = \operatorname{epi}(v) \cap \{(t,x) \mid t \leq u_k^*\},$$

Also, let $\mathcal{K}$ be a unit circular cone $\mathcal{K} = \operatorname{conv}\big\{\{(1,x) \mid \frac{1}{2}\|x\|^2 \leq 1\}, 0\big\}$. First note that $\mathcal{K}_k$ and $\mathcal{K}$, our standard cone, are related by an affine transformation. This transform which first involves the translation of the tip of the cones to $(0,0)$, followed by a scaling:

$$\mathcal{K} = \Sigma_k^{-1}(\mathcal{K}_k - (\min_{x\in\mathcal{X}} f(x), x^*)) \tag{7.2.1}$$

$$\Sigma_k = \operatorname{diag}([u_k - \min_{x\in\mathcal{X}} f(x), ([u_k - \min_{x\in\mathcal{X}} f(x)]/L)^{1/2}, \ldots, ([u_k - \min_{x\in\mathcal{X}} f(x)/L])^{1/2}). \tag{7.2.2}$$

Therefore, we have following chain of inequalities:

$$\operatorname{size}(P_k) \overset{(a)}{\geq} \operatorname{size}(\mathcal{P}_k) \overset{(b)}{\geq} \operatorname{size}(\mathcal{K}_k) \overset{(c)}{=} \det(\Sigma_k)\operatorname{size}(\mathcal{K}) = \operatorname{size}(\mathcal{K})[u_k - \min_{x\in\mathcal{X}} f(x)]^{(2+n)/2}/L^{n/2},$$

where (a) comes from Assumption (7.1.1) and (7.1.2), which implies $P_k \subseteq \mathcal{P}_k$, (b) follows from the fact that $\mathcal{P}_k \supseteq \mathcal{K}_k$ by construction ($\mathcal{K}_k$ is a circular cone inscribed in the parabola $\mathcal{P}_k$), (c) comes from translating and scaling $\mathcal{K}_k$ into the standard cone, cf. (7.2.1), the affine invariance of the MIE, and the affine homogeneity of size. Taking powers of $2/(2+n)$ on both sides, we get the final result:

$$u_k^* - \min_{x\in\mathcal{X}} f(x) \leq \frac{L^{n/(2+n)}\operatorname{size}(P_k)^{2/(2+n)}}{\operatorname{size}(\mathcal{K})^{2/(2+n)}} \leq \mathcal{O}(1)\cdot n[L^n \operatorname{size}(P_k)^2]^{1/(2+n)}.$$

For the final inequality, we use the fact that

$$\operatorname{size}(\mathcal{K})^{2/(2+n)} \geq \omega_n^{2/(2+n)}/4^{\frac{2n+2}{2+n}} \geq n\mathcal{O}(1)$$

for all n. $\square$

This theorem can be seen as a quadratic improvement over Lemma 4.3.1 in which we proved $u_k^* - \min_{x\in\mathcal{X}} f(x) \lessapprox \operatorname{size}(P_k)^{1/(n+1)}$. This theorem improves that bound to $u_k^* - \min_{x\in\mathcal{X}} f(x) \lessapprox \operatorname{size}(P_k)^{2/(2+n)}$. This fact allows us to relax the termination condition by a squared factor. Note too, that this improvement does not depend on the parabolic cutting planes.

Though the above lemma does not change the algorithm's complexity class, its utility will be seen when used in concert with the next result. The next theorem establishes a lower bound on ϵ_k

as a function of size(P_k). The parabolic lower bounds, intuitively, prevent P_k from becoming too flat.

Lemma 7.2.2. (Relating ϵ_k and size(P_k)**)** For all k

$$\epsilon_k \geq \mathcal{O}(1) \cdot \gamma[\mu^n \text{size}(P_k)^2]^{1/(2+n)},$$

where $\mathcal{O}(1)$ is a universal constant.

Proof. Let l_k^* be the strongest lower bound at iteration k, i.e.,

$$l_k^* = \max_{i<k}\{l_i + g_i^T(\cdot - x_i) + \tfrac{\mu}{2}\|\cdot - x_i\|^2\}.$$

Since each function within the max is strongly convex with modulus μ, so must l_k^* (strong convexity is preserved by the max). Let $\hat{x}^*$ be the minimizer of l_k^*. Since $0 \in \partial l_k^*(\hat{x}^*)$ (see Figure 7.2.1),

$$l_k^*(x) \geq l_k^*(\hat{x}^*) + 0^T(x - \hat{x}^*) + \tfrac{\mu}{2}\|x - \hat{x}^*\|^2 \geq l_k^*(\hat{x}^*) + \tfrac{\mu}{2}\|x - \hat{x}^*\|^2.$$

This implies $\text{epi}(l_k^*) \subseteq \text{epi}(l_k^*(\hat{x}^*) + \frac{\mu}{2}\|\cdot - \hat{x}^*\|^2)$ and therefore that

$$P_k = \text{epi}(l_k^*) \cap \{(t, x) \mid t \leq u_k^*\} \tag{7.2.3}$$

$$\subseteq \text{epi}(l_k^*(\hat{x}^*) + \tfrac{\mu}{2}\|\cdot - \hat{x}^*\|^2) \cap \{(t, x) \mid t \leq u_k^*\} =: \mathcal{P}_k. \tag{7.2.4}$$

See Figure 7.2.1 for an illustration. Let ω_n be the volume of a unit n-ball, so that

$$\begin{aligned}
\text{size}(P_k) &\overset{(a)}{\leq} \text{vol}(P_k) \\
&\overset{(b)}{\leq} \text{vol}(\mathcal{P}_k) \\
&\overset{(c)}{=} \frac{2\omega_n}{n[\mu/2]^{n/2}} \cdot \text{height}(\mathcal{P}_k)^{(2+n)/2} \\
&\overset{(d)}{=} \frac{2\omega_n}{n[\mu/2]^{n/2}} \cdot \text{height}(P_k)^{(2+n)/2} \overset{(e)}{\leq} \frac{2\omega_n}{n[\mu/2]^{n/2}} \cdot [\gamma^{-1}(n+1)\epsilon_k]^{(2+n)/2}.
\end{aligned}$$

Inequality (a) comes from the fact that size$(P_k) = \text{vol}(\mathcal{E}_k)$, and $\mathcal{E}_k$, being an inscribed ellipse, is a subset of P_k. Inequality (b) comes from (7.2.3). In equality (c) we use the volume of a parabola

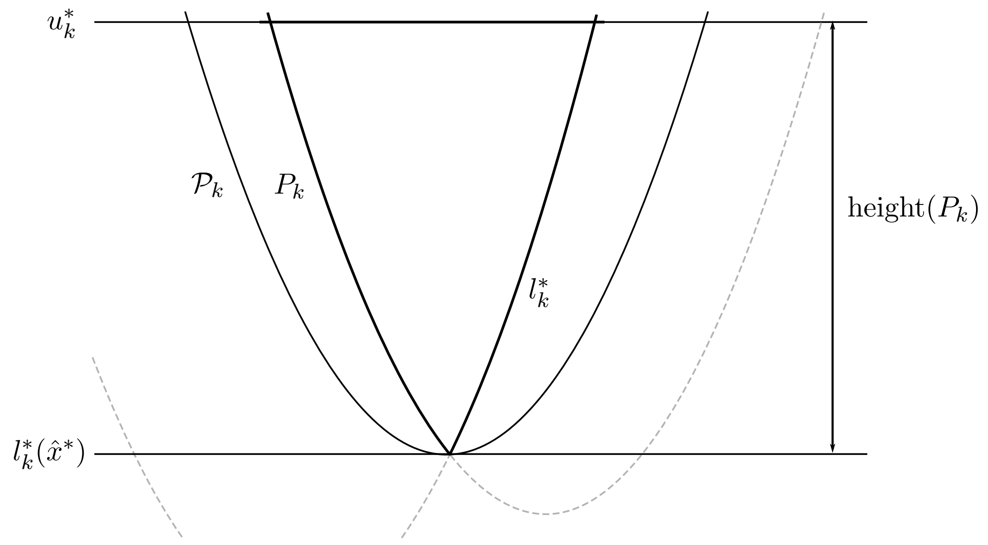


FIGURE 7.2.1. Geometric illustration of the parabolic outer approximation of P_k, i.e., $P_k \subseteq \mathcal{P}_k$.

(see Figure 7.2.1), and equality (d) comes from the fact by construction height$(\mathcal{P}_k) =$ height(P_k). The final inequality (e) comes from Lemma 6.2.1.

Taking powers of $2/(2+n)$ on both sides, we obtain

$$\epsilon_k \geq \frac{\gamma n^{2/(2+n)}}{2} \cdot \frac{[\text{size}(P_k)^2 \mu^n]^{1/(2+n)}}{(n+1)[\omega_n]^{2/(2+n)}} \geq \mathcal{O}(1) \cdot \gamma[\mu^n \text{size}(P_k)^2]^{1/(2+n)}.$$

The final inequality comes from Lemma A.1.3. □

Finally, we bound the total amount of effort involved in the algorithm. This is done in an unconventional way: instead of bounding the total amount of work at each iteration, as is the conventional approach, we must work backwards from the terminus to bound the total effort involved.

Lemma 7.2.3. (**Maximization of a convex function over a cone**) Let f be a convex function, and A be an invertible matrix. Furthermore assume that f is bounded on the cone $\{x \mid Ax \geq b\}$. Then $\sup_x \{f(x) \mid Ax \geq b\} = f(A^{-1}b)$.

PROOF. We transform the problem to

$$\sup_x \{f(x) \mid Ax \geq b\} \overset{(a)}{=} \sup_x \{f(A^{-1}(y+b)) \mid y \geq 0\} = \sup_y \{g(y) \mid y \geq 0\},$$

where $g(y) = f(A^{-1}(y+b))$. Equality (a) comes from a change of variables, $y = Ax - b$.

By assumption, the function g is bounded, and therefore is decreasing on every half line [Roc70], and achieves the maximum at $g(0)$. Therefore, g must attain a global minimum at $g(0) = f(A^{-1}b)$. □

We assume that the amount of work is inversely proportional to ϵ_{tol}.

Theorem 7.2.4. (Total effort, parabolic inexact epigraph method of inscribed ellipsoids) Assume the algorithm is run till termination. Then for $\epsilon_{\text{tol}} > 0$

$$\sum \frac{1}{\epsilon_k} \le \frac{L}{\mu} \cdot \frac{\mathcal{O}(1)}{\gamma(1 - [0.843/(1-\gamma)^2]^{2/(2+n)})} \cdot \frac{n}{\epsilon_{\text{tol}}},$$

where $\mathcal{O}(1)$ represents a universal constant.

Proof. Let $\eta_k := \text{size}(P_k)^{2/(2+n)}$. Then we have the following three guarantees:

$$\begin{aligned} &\epsilon_k \ge C_1 \eta_k && \text{(Lower bound on } \epsilon_k\text{, Lemma 7.2.2)} \\ &\eta_k \ge C_2 \epsilon_{\text{tol}} && \text{(Termination criteria, Algorithm 7.1)} \\ &\eta_k \ge (0.843/(1-\gamma))^{-2/(2+n)} \cdot \eta_{k+1} && \text{(Convergence guarantee, Lemma 6.2.4)} \end{aligned}$$

where

$$C_1 = \mathcal{O}(1) \cdot \mu^{n/(2+n)}, \qquad C_2 = \mathcal{O}(1) \cdot L^{-n/(2+n)}.$$

Therefore, the total cost of the iteration is

$$\sum \frac{1}{\epsilon_k} \le \sup\{\alpha_0, \alpha_1, \dots\},$$

where α_j is the total amount of work performed if the algorithm ran for j steps. We can upper bound the total amount of work performed by performing an optimization over the three constraints described above. Let $\beta = [0.843/(1-\gamma)^2]^{-2/(2+n)}$. Then

$$\begin{aligned} \alpha_j &\le \sup\{\textstyle\sum_{k=0}^{j} \epsilon_k^{-1} \mid \eta_0 \ge \beta\eta_1, \cdots, \eta_{j-1} \ge \beta\eta_j \text{ and } \eta_k \ge C_2\epsilon_{\text{tol}}, \epsilon_k \ge C_1\eta_k, \forall k\} \\ &= \sup\{\textstyle\sum_{k=0}^{j} \epsilon_k^{-1} \mid \eta_0 \ge \beta\eta_1, \cdots, \eta_{j-1} \ge \beta\eta_j \text{ and } \eta_k \ge C_2\epsilon_{\text{tol}}, \epsilon_k = C_1\eta_k, \forall k\} \end{aligned}$$

$$= \sup \{ \textstyle\sum_{k=0}^{j} [C_1 \eta_k]^{-1} \mid \eta_0 \geq \beta\eta_1, \cdots, \eta_{j-1} \geq \beta\eta_j, \eta_j \geq C_2 \epsilon_{\text{tol}} \}$$

$$\overset{(a)}{=} \frac{\sum_{k=0}^{j} \beta^{-k}}{C_1 C_2 \epsilon_{\text{tol}}} \leq \frac{\sum_{k=0}^{\infty} \beta^{-k}}{C_1 C_2 \epsilon_{\text{tol}}} = \frac{1}{(1-\beta^{-1}) C_1 C_2 \epsilon_{\text{tol}}}.$$

Inequality (a) comes from Lemma 7.2.3. Our objective function is convex, and bounded from above on the feasible set (the function is decreasing, and the feasible set does not contain 0). Let us write the constraints in the following vectorized form:

$$\begin{pmatrix} 1 & -\beta & & \\ & \ddots & \ddots & \\ & & 1 & -\beta \\ & & & 1 \end{pmatrix} \begin{pmatrix} \eta_0 \\ \vdots \\ \eta_{j-1} \\ \eta_j \end{pmatrix} \geq \begin{pmatrix} 0 \\ \vdots \\ 0 \\ C_2 \epsilon \end{pmatrix},$$

and observe that Lemma (7.2.3) states that the optimum is achieved at equality. Therefore by an elementary process of back-substitution, it is clear the optimum is achieved at $\text{size}_k = \beta^{j-k} C_2 \epsilon_{\text{tol}}$. Substituting this back into the objective and reversing the order of summation yields the equality. Finally,

$$\frac{1}{C_1 C_2} = \frac{\mathcal{O}(1) \cdot n L^{n/(2+n)}}{\mathcal{O}(1) \cdot \gamma \mu^{n/(2+n)}} \leq \mathcal{O}(1) \cdot \frac{nL}{\gamma\mu},$$

because $L/\mu \geq 1$ which implies $(L/\mu)^p \leq L/\mu$ for $p \leq 1$. $\square$

Note that we can optimize the upper bound with respect to γ to get the best choice of γ. We investigate this numerically—generally, the larger the dimension, the smaller the γ we should use, but this approaches 0.04137 approximately.

Surprisingly, there are no constants in the above proof which involve size(P_0). This counterintuitive fact means the volume of the initial localization polytope plays no role in the optimization. How can this be possible? Perhaps this can be understood with the following intuition. Since the iterates at the beginning get exponentially cheaper as the polytope P_0 increases, in a sense, early iterations contribute very little to the work involved. In practice, of course, this is not the case. There is some, finite fixed cost in solving any optimization problem. However we believe that this property still translates to much better performance in practice.

Part 4

Proximal methods

CHAPTER 8

Proximal gradient

While cutting plane algorithms work by elimination, proximal gradient methods work by exploration. The strategy is greedy—given a current iterate x_k, proximal gradient methods use local information to find the most promising next point, and moves there, securing a small, local amount of decrease at each step. The extent to which local information is "good" determines how fast we converge, and hence the convergence of proximal gradient methods does not depend on the dimension of the problem, but how well behaved the function is itself. Recall that proximal gradient methods operate on problems with composite structure (1.3.1),

$$\underset{x}{\text{minimize}} \qquad f(x) + g(x)$$

where the functions $f : \mathbf{R} \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ are convex. We assume that f has a Lipschitz-continuous gradient, and that g is lower semicontinuous [Roc70, Definition 1.5]. Recall the scaled proximal iteration, defined in equation (1.3.2), is

$$x_{k+1} = \mathbf{prox}_g^{H_k}(x_k - H_k^{-1}\nabla f(x_k)),$$

for some sequence of curvature approximations H_k. The most popular variation of this algorithm is the choice of $H_k = \alpha_k I$. The popularity of this method can be attributed to a synergy of the broad base of applications for problems with composite structure, and abundance of cheap proximal operators which make the computation of the proximal operator efficient.

8.1. Problems with composite form

We consider two classes of applications of composite structure, primal and dual methods. In a typical primal applications, f is a smooth loss function that penalizes incorrect predictions of a model, and g is a regularizer that encourages desirable structure in the solution. One can pick any smooth loss from Section 3.1.1 and combine it with any regularizer from Section 3.1.2. A canonical example of this is the sparse least squares problem

$$\underset{x}{\text{minimize}} \qquad \underbrace{\tfrac{1}{2}\|Ax-b\|^2}_{f(x)} + \underbrace{\lambda\|x\|_1}_{g(x)},$$

which has applications in compressed sensing, see Section 3.1.3, and feature selection, see Section 3.1.2.2. The accelerated variation of proximal gradient is the celebrated iterative shrinkage algorithm by Becker and Teboulle [BT09].

When the loss function is non-smooth, but the regularizer has a smooth conjugate, we can reverse the roles of regularizer and loss via Fenchel duality to have it conform to problem (1.3.1). Observe that in both regression and classification, the loss function is a separable function of Ax, where A is defined in (3.1.2), and hence we can write it as

$$\underset{x}{\text{minimize}} \qquad f_1(Ax) + f_2(x) \qquad \text{with dual} \qquad \underset{v}{\text{minimize}} \qquad f_1^*(v) + f_2^*(A^Tv).$$

The support vector machine problem in Example 5.2.5, for example, requires the minimization

$$\underset{x}{\text{minimize}} \qquad \mathbf{1}^T\max\{0, 1-BAx\} + \tfrac{1}{2}\|x\|^2 \qquad\qquad (\mathbf{P}\text{-SVM}).$$

We have the correspondence $f_1(y) = \mathbf{1}^T\max\{0, 1-By\}$ with conjugate $f_1^*(v) = \delta(B^{-1}x|[0,1]^m) - \mathbf{1}^TB^{-1}v$ and $f_2(x) = f_2^*(x) = \frac{1}{2}\|x\|^2$. This gives us the dual which corresponds to a box constrained quadratic program,

$$\underset{x}{\text{minimize}} \qquad \underbrace{\tfrac{1}{2}\|A^Tv\|^2 - \mathbf{1}^TB^{-1}v}_{f(x)} + \underbrace{\delta(B^{-1}v|[0,1]^m)}_{g(x)} \qquad\qquad (\mathbf{D}\text{-SVM}),$$

which can be solved by proximal gradient. Given an optimal solution y^*, the primal can be recovered by $x^* = -A^Ty^*$. The case of robust loss too admits a similar construction, with the primal-dual pairings $f_1(y) = \|y-b\|_1, f_2(v) = \delta(v|[-1,1]^m) - b^Tv$ and $f_1^* = f_2^* = \frac{1}{2}\|\cdot\|^2$, resulting too in a dual equal to a box constrained quadratic program, that can be recovered with an identical formula.

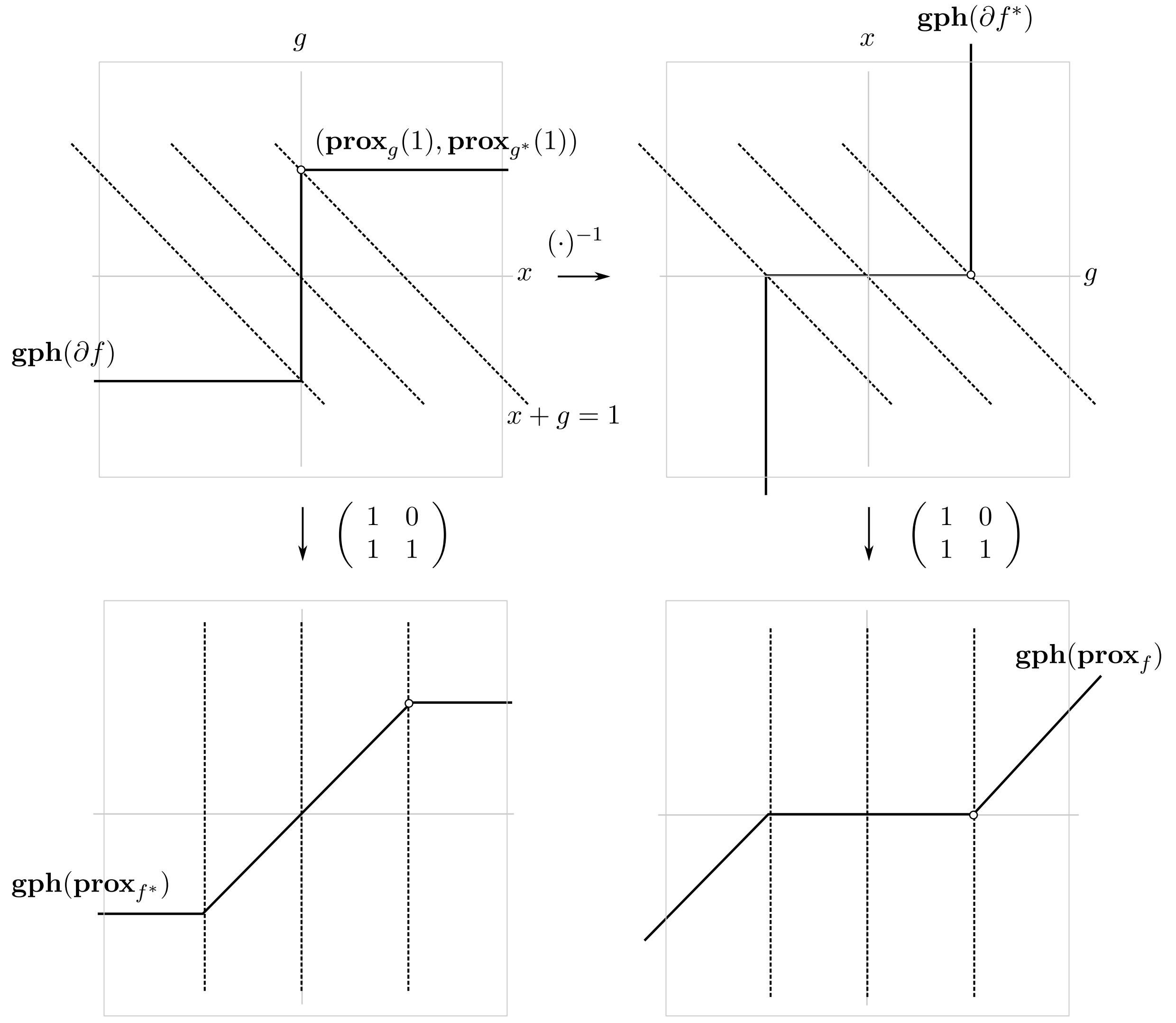

g
x
$\mathbf{gph}(\partial f^*)$
$(\mathbf{prox}_g(1), \mathbf{prox}_{g^*}(1))$
$(\cdot)^{-1}$
x
g
$\mathbf{gph}(\partial f)$
$x + g = 1$
$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
$\mathbf{gph}(\mathbf{prox}_f)$
$\mathbf{gph}(\mathbf{prox}_{f^*})$

FIGURE 8.2.1. Visualizing the $\mathbf{prox}_g$ where $g = |\cdot|$

8.2. Computing the proximal operator

For many g's of interest, the proximal operator $\mathbf{prox}_g$ in equation (2.2.2) can be computed efficiently. This generalizes, via a rescaling of the coordinates, to the scaled proximal operator with diagonal scaling matrix, but we omit this more general version for clarity. We consider two classes of g's which admit easy proximal computations.

8.2.0.1. *Separable functions.* When g is separable, the problem decomposes into n one-dimensional optimization problems, which can often be computed in closed form by elementary methods, and in the worst case can be evaluated numerically using a root finding algorithm such as Netwon's method.

In a single dimension, the proximal operator has a clear visual interpretation. Define the graph of the set valued subgradient map to be [RW98, Chapter 5]

$$\mathbf{gph}(\partial f) := \{(g,x) \mid g \in \partial f(x)\}\,.$$

Then the graphs of $\mathbf{prox}_g$ have the following relation

$$\mathbf{gph}(x - \mathbf{prox}_f) = \mathbf{gph}(\mathbf{prox}_{f^*}) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \mathbf{gph}(\partial f).$$

In other words, the point in which the line $\{(g,x) \mid x + g = z\}$ intersects the graph of ∂g is exactly $(\mathbf{prox}_f(z), \mathbf{prox}_{f^*}(z))$—this is the Minty parametrization of the subgradient map. We illustrate this in Figure 8.2.1 to derive the proximal operator for $g(x) = |x|$ and $g^*(x) = \delta(x|[-1,1])$ visually.

8.2.0.2. *Homogeneous functions.* The set of proper closed convex homogeneous functions, $g(\lambda x) = \lambda g(x)$, are also the set of functions for which conjugate admits a support function representation [Roc70, Corollary 13.2.1],

$$g(x) = \sup_{v\in S}\{v^T x\} = \delta(\cdot|S)^*(x)$$

for some convex set S. Thus we can reduce the computation of the proximal operator to the projection via Moreau's identity, equation (2.3.2), to recover the proximal operator of g as $x - \mathbf{prox}_{\delta(\cdot|S)}(x)$. The 2-norm, for example, is the support function of the ball $B_2 = \{x \mid \|x\| \le 1\}$ and hence

$$\mathbf{prox}_{\|\cdot\|_1}(x) = x - \mathbf{prox}_{\delta(\cdot,B_2)}(x) = x - \begin{cases} x & \text{if } \|x\| \le 1 \\ x/\|x\| & \text{otherwise.} \end{cases}$$

and the proximal operator of the infinity norm by the projection onto the 1-norm ball, computable in linear time by [DSSSC08].

These tricks do not generalize readily to the scaled proximal operator where H is dense. In the next section we propose a different strategy for computing the scaled proximal operator.

CHAPTER 9

Efficient evaluation of the scaled proximal operator

In this section we will show an important family of functions g and H, for which the scaled proximal operator (1.3.6) can be computed efficiently via an interior method. This approach builds on the work of [ABP13], who define the class of quadratic-support functions and outline a particular interior algorithm for their optimization. Our approach is specialized to the case where the quadratic-support function appears inside of a proximal computation. Together with the correct dualization approach (§9.3), this yields a particularly efficient interior implementation when the data that define g and H have special structure (§9.4).

9.1. Quadratic-support functions

Aravkin et al. [ABP13] introduced the notion of a quadratic-support (QS) function, a generalization of sublinear support functions [RW98, Ch. 8E]—here we introduce a slightly more general definition than the version implemented by Aravkin et al. We retain the “QS” designation because the quadratic term, which is an essential feature of their definition, can also be expressed by the version we use here.

Let $\mathbf{B}_p = \{z \mid \|z\|_p \le 1\}$ and $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_k$, where each cone $\mathcal{K}_i$ is either a nonnegative orthant $\mathbf{R}^m_+$ or a second-order cone introduced in (2.1.2). (The size m of the cones may of course be different for each index i.) Recall that $Ay \succeq_{\mathcal{K}} b$ means that $Ay - b \in \mathcal{K}$, (2.1.1), and in this section, unless otherwise specified, x is an n-vector. To help disambiguate dimensions, the p-by-p identity matrix is denoted by I_p, and the p-vector of all ones by $\mathbf{1}_p$.

We consider the class of functions $g : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ that have the conjugate representation

$$g(x) = \sup_y \{y^T(Bx + d) \mid y \in \mathcal{Y}\}, \quad \text{where} \quad \mathcal{Y} = \{y \in \mathbf{R}^\ell \mid Ay \succeq_{\mathcal{K}} b\}. \tag{9.1.1}$$

(The term "conjugate" alludes to the implicit duality, since g may be considered as the conjugate of the indicator function to the set $\mathcal{Y}$.) We assume throughout that the feasible set $\mathcal{Y}$ is nonempty. If $\mathcal{Y}$ contains the origin, the QS function g is nonnegative for all x, and we can then consider it to be a penalty function. This is automatically true, for example, if $b \leq 0$.

The formulation (9.1.1) is close to the standard definition of a sublinear support function [Roc70, §13], which is recovered by setting $d = 0$ and $B = I$, and letting $\mathcal{Y}$ be any convex set. Unlike a standard support function, g is not positively homogeneous if $d \neq 0$. This is a feature that allows us to capture important classes of penalty functions that are not positively homogeneous, such as piecewise quadratic functions, the "deadzone" penalty, or indicators on certain constraint sets. These are examples that are not representable by the standard definition of a support function. Our definition springs from the quadratic-support function definition introduced by [ABP13], who additionally allow for an explicit quadratic term in the objective and for $\mathcal{Y}$ to be any nonempty convex set. The concrete implementation considered by [ABP13], however, is restricted to the case where $\mathcal{Y}$ is polyhedral. In contrast, we also allow $\mathcal{K}$ to contain second-order cones. Therefore, any quadratic objective terms in the [ABP13] definition can be "lifted"and turned into a linear term with a second-order-cone constraint (see Example 9.1.2). Our definition is thus no less general.

This expressive class of functions includes many penalty functions commonly used in machine learning; [ABP13] give many other examples. In addition, they show how to interpret QS functions as the negative log of an associated probability density, which makes these functions relevant to maximum a posteriori estimation. In the remainder of this section we provide some examples that illustrate various regularizing functions and constraints that can be expressed as QS functions.

Example 9.1.1 (1-norm regularizer)**.** The 1-norm has the QS representation

$$\|x\|_1 = \sup_y \{\, y^T x \mid y \in \mathbf{B}_\infty \,\},$$

where

$$A = \begin{pmatrix} I_n \\ -I_n \end{pmatrix}, \quad b = -\begin{pmatrix} \mathbf{1}_n \\ \mathbf{1}_n \end{pmatrix}, \quad d = 0, \quad B = I_n, \quad \mathcal{K} = \mathbf{R}^{2n}_{+}. \tag{9.1.2}$$

Example 9.1.2 (2-norm)**.** This simple example illustrates how the QS representation (9.1.1) can represent the 2-norm, which is not possible using the QS formulation described by [ABP13] where the constraints are polyhedral. With our definition, the 2-norm has the QS representation

$$\|x\|_2 = \sup_y \{y^T x \mid y \in \mathbf{B}_2\} = \sup_y \{y^T x \mid (1, y) \succeq_{\mathcal{K}} 0\},$$

where

$$A = \begin{pmatrix} 0 \\ I_n \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad d = 0, \quad B = I_n, \quad \mathcal{K} = \mathcal{Q}^{n+1}. \tag{9.1.3}$$

Example 9.1.3 (Polyhedral norms)**.** Any polyhedral seminorm is a support function, e.g., $\|Bx\|_1$ for some matrix B. In particular, if the set $\{y|Ay \ge b\}$ contains the origin and is centro-symmetric, then

$$\|x\| := \sup_y \{y^T Bx \mid Ay \ge b\}$$

defines a norm if B is nonsingular, and a seminorm otherwise. This is a QS function with $d := 0$ (as will be the case for any positively homogeneous QS function) and $\mathcal{Y} := \{y \mid Ay \ge b\}$.

Example 9.1.4 (Quadratic function)**.** This example justifies the term "quadratic" in our modified definition, even though there are no explicit quadratic terms. It also illustrates the roles of the terms B and d. The quadratic function can be written as

$$\tfrac{1}{2}\|x\|_2^2 = \sup_{y,\,t} \{y^T x - \tfrac{1}{2} t \mid \|y\|_2^2 \le t\} = \sup_{y,\,t} \left\{ \begin{pmatrix} y \\ t \end{pmatrix}^T \left[\begin{pmatrix} I_n \\ 0 \end{pmatrix} x - \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} \right] \,\middle|\, \|y\|_2^2 \le t \right\}.$$

Use the derivation in C.1 to obtain the QS representation with parameters

$$A = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ I_n & 0 \end{pmatrix}, \quad b = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 0 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}, \quad B = \begin{pmatrix} I_n \\ 0 \end{pmatrix}, \quad \mathcal{K} = \mathcal{Q}^{n+2}.$$

Example 9.1.5 (1-norm indicator)**.** This example is closely related to the 1-norm regularizer in Example 9.1.1, except that the QS function is used to express the constraint $\|x\|_1 \le 1$ via an indicator function $g = \delta(\,\cdot \mid \mathbf{B}_1)$. Write the indicator to the 1-norm ball as the conjugate of the infinity norm, which gives

(9.1.4)
$$\delta(x \mid \mathbf{B}_1) = \sup_y \{y^T x - \|y\|_\infty\} = \sup_{y,\tau}\{y^T x - \tau \mid y \in \tau \mathbf{B}_\infty\} = \sup_{y,\tau}\{y^T x - \tau \mid -\tau \mathbf{1}_n \le y \le \tau \mathbf{1}_n\}.$$

This is a QS function with parameters

$$A = \begin{pmatrix} -I_n & \mathbf{1}_n \\ I_n & \mathbf{1}_n \end{pmatrix}, \quad b = 0, \quad d = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad B = \begin{pmatrix} I_n \\ 0 \end{pmatrix}, \quad \mathcal{K} = \mathbf{R}^{2n}.$$

Example 9.1.6 (Indicators on polyhedral cones)**.** Consider the following polyhedral cone and its polar:

$$U = \{x \mid Bx \le 0\} \quad \text{and} \quad U^\circ = \{B^T y \mid y \ge 0\}.$$

Use the support-function representation of a cone in terms of its polar to obtain

$$\delta(x \mid U) = \delta^*(\cdot \mid U^\circ)(x) = \sup_y \{y^T Bx \mid y \ge 0\}, \tag{9.1.5}$$

which is an example of an elementary QS function. A concrete example is the positive orthant, obtained by choosing $B = I_n$. An important example, used in isotonic regression [BC90], is the monotonic cone

$$U := \{x \mid x_i \ge x_j, \ \forall (i,j) \in \mathbf{E}\},$$

Here, $\mathbf{E}$ is the set of edges in a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ that describes the relationships between variables in $\mathbf{V}$. If we set B to be the incidence matrix for the graph, (9.1.5) then corresponds to the indicator on the monotonic cone U.

Example 9.1.7 (Distance to a cone)**.** The distance to a cone U that is a combination of polyhedral and second-order cones can be represented as a QS function:

$$\begin{aligned} \inf_{x \in U} \|x - y\|_2 &= \inf_x \{\|x - y\|_2 + \delta(x \mid U)\} \\ &= \left[\delta(\cdot \mid \mathbf{B}_2) + \delta(\cdot \mid U^\circ)\right]^*(y) = \ \sup \left\{y^T x \mid y \in \mathbf{B}_2 \cap U^\circ\right\}. \end{aligned}$$

The second equality follows from the relationship between infimal convolution and conjugates [Roc70, §16.4]. When U is the positive orthant, for example, $g(x) = \|\max\{0,\, x\}\|_2$, where the *max* operator is taken elementwise.

9.2. Building quadratic-support functions

Quadratic-support functions are closed under addition, composition with an affine map, and infimal convolution with a quadratic function. In the following, let g_i be QS functions with parameters A_i, b_i, d_i, B_i, and $\mathcal{K}_i$ (with $i = 0, 1, 2$). The rules for addition and composition are described in [ABP13], which are here summarized and amplified.

Addition rule: The function $h(x) := g_1(x) + g_2(x)$ is QS with parameters

$$A = \begin{pmatrix} A_1 & \\ & A_2 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad \mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2.$$

Concatenation rule: The function $h(x) := g_0(x^1) + \cdots + g_0(x^k)$, where each partition $x^i \in \mathbf{R}^n$, is QS with parameters

$$(A,\, B) = I_k \otimes (A_0,\, B_0), \qquad (b,\, d) = \mathbf{1}_k \otimes (b_0,\, d_0), \qquad \mathcal{K} = \mathcal{K}_0 \times \overset{(k)}{\cdots} \times \mathcal{K}_0. \tag{9.2.1}$$

where the symbol $\otimes$ denotes the Kronecker product. The rule for concatenation follows from the rule for addition of QS functions.

Affine composition rule: The function $h(x) := g_0(Px - p)$ is QS with parameters

$$A = A_0, \quad b = b_0, \quad d = d_0 - B_0 p, \quad B = B_0 P.$$

Moreau-Yosida regularization: The Moreau-Yosida envelope of g_0 is the value of the proximal operator, i.e., $\mathbf{env}^H_{g_0}(z) := \inf_x\{\frac{1}{2}\|z - x\|^2_H + g_0(x)\}$. It follows from [BH13, Proposition 4.10] that

$$\mathbf{env}^H_{g_0}(z) = \sup_y \left\{ y^T(B_0 x + d_0) - \frac{1}{2} y^T B_0 H^{-1} B_0^T y \mid A_0 y \succeq_{\mathcal{K}_0} b_0 \right\},$$

which is a QS function with parameters

$$A = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ R & 0 \\ A_0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 0 \\ b_0 \end{pmatrix}, \quad d = \begin{pmatrix} d_0 \\ -\frac{1}{2} \end{pmatrix}, \quad B = \begin{pmatrix} B_0 \\ 0 \end{pmatrix}, \quad \mathcal{K} = \mathcal{Q}^{n+2} \times \mathcal{K}_0,$$

with Cholesky factorization $R^T R = B_0 H^{-1} B_0^T$. The derivation is given in Appendix C.1, where we take $Q = B_0 H^{-1} B_0^T$.

Example 9.2.1 (Sums of norms)**.** In applications of group sparsity, equation (3.1.16), various norms are applied to all partitions of $x = (x^1, \dots, x^p)$, which possibly overlap. Consider the case of adding two norms $g(x) = \|x^1\|_\triangledown + \|x^2\|_\vartriangle$. (The extension to adding three or more norms follows trivially.) First, we introduce matrices P_i that restrict x to partition i, i.e., $x^i = P_i x$, for $i = 1, 2$. Then

$$g(x) = \|P_1 x\|_\triangledown + \|P_2 x\|_\vartriangle.$$

Then we apply the affine-composition and addition rules to determine the corresponding quantities that define the QS representation of g:

$$A = \begin{pmatrix} A_\triangledown & \\ & A_\vartriangle \end{pmatrix}, \quad b = \begin{pmatrix} b_\triangledown \\ b_\vartriangle \end{pmatrix}, \quad d = 0, \quad B = \begin{pmatrix} B_\triangledown P_1 \\ B_\vartriangle P_2 \end{pmatrix}, \quad \mathcal{K} = \mathcal{K}_\triangledown \times \mathcal{K}_\vartriangle,$$

where A_i, b_i, B_i, and $\mathcal{K}_i$ (with $i = \triangledown, \vartriangle$) are the quantities that define the QS representation of the individual norms. (Necessarily, $d = 0$ because the result is a norm and therefore positive homogeneous.) In the special case where $\|\cdot\|_\triangledown$ and $\|\cdot\|_\vartriangle$ are both the 2-norm, then A_i, b_i, B_i, and $\mathcal{K}_i$ are given by (9.1.3) in Example 9.1.2.

Example 9.2.2 (Graph-based 1-norm and total variation)**.** A variation of Example 9.2.1 can be used to define the total variation regularizers discussed in see Section 9.7.2.4. Let

$$g(x) = \|Nx\|_G \quad \text{with} \quad \|z\|_G = \sum_{i=1}^{p} \|z^i\|_2,$$

where z^i is a partition of z and N is an m-by-n matrix. For anisotropic TV and the graph-based 1-norm regularizer, N is the adjacency matrix of a graph, and each partition z^i has a single unique element, so $g(x) = \|Nx\|_1$. For isotropic TV, each partition captures neighboring pairs of variables, and N is a finite-difference matrix. The QS addition and affine-composition rules can be combined to derive the parameters of g. When $p = m$ (i.e., each z^i is a scalar), we are summing n absolute-value functions, and we use (9.1.2) and (9.2.1) to obtain

$$A = I_m \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad b = \mathbf{1}_m \otimes \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad d = 0, \quad B = N, \quad \mathcal{K} = \mathbf{R}^{2m}_+. \tag{9.2.2}$$

Now consider the variation where $p = m/2$, (i.e., each partition has size 2), which corresponds to summing $m/2$ two-dimensional 2-norms. Use (9.1.3) to obtain

$$A = I_{m/2} \otimes \begin{pmatrix} 0 \\ I_2 \end{pmatrix}, \quad b = \mathbf{1}_{m/2} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad d = 0, \quad B = N, \quad \mathcal{K} = \mathcal{Q}^2 \times \overset{(m/2)}{\cdots} \times \mathcal{Q}^2.$$

9.3. The proximal operator as a conic QP

We describe in this section how the scaled proximal map (1.3.6) can be obtained as the solution of a quadratic optimization problem (QP) over conic constraints,

$$\underset{y}{\text{minimize}} \quad \tfrac{1}{2} y^T Q y - c^T y \quad \text{such that} \quad Ay \succeq_{\mathcal{K}} b, \tag{9.3.1}$$

for some positive semidefinite ℓ-by-ℓ matrix Q and a convex cone $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_k$. The transformation to a conic QP is not immediate because the definition of the QS function implies that the proximal map involves nested optimization. Duality, however, furnishes a means for simplifying this problem.

Proposition 9.3.1. Let g be a QS function. The following problems are dual pairs:

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|z - x\|_H^2 + g(x), \tag{9.3.2a}$$

$$\underset{Ay \succeq_{\mathcal{K}} b}{\text{minimize}} \quad \tfrac{1}{2} y^T B H^{-1} B^T y - (d + Bz)^T y. \tag{9.3.2b}$$

If strong duality holds, the corresponding primal-dual solutions are related by

$$Hx + B^T y = Hz. \tag{9.3.3}$$

Proof. Let

$$h_1(x) := \tfrac{1}{2}\|x - z\|_H^2 \quad \text{and} \quad h_2(x) := \sup_{y \in \mathcal{Y}} \{y^T(x + d)\}.$$

If strong duality holds, it follows from Fenchel duality, Theorem (2.3.1) that

$$\inf_x \{h_1(x) + h_2(Bx)\} = -\inf_y \{h_1^*(-B^T y) + h_2^*(y)\}, \tag{9.3.4}$$

where

$$h_1^*(y) = \tfrac{1}{2}\|y\|_{H^{-1}}^2 + z^T y \quad \text{and} \quad h_2^*(y) = \delta(z \mid \mathcal{Y}) - d^T y$$

are the Fenchel conjugates of h_1 and h_2, and the infima on both sides are attained. (See [Roc70, §12] for the convex calculus of Fenchel conjugates.) The right-hand side of (9.3.4) is precisely the dual problem (9.3.2b). It also follows from Fenchel duality that the pair (x, y) is optimal only if

$$x \in \operatorname{argmin}_x \{h_1(x) + y^T Bx\}.$$

Differentiate this objective to obtain (9.3.3). □

Strong duality holds when $B \cdot \operatorname{ri} \operatorname{dom}(h_1) \cap \operatorname{ri} \operatorname{dom}(h_2) \neq \emptyset$. This holds, for example, when the interior of the domain of g is nonempty, since

$$\operatorname{int} \operatorname{dom}(g) \neq \emptyset \iff \operatorname{im} B \cap \operatorname{int} \operatorname{dom}(h_2) \neq \emptyset \Rightarrow B \cdot \operatorname{ri} \operatorname{dom}(h_1) \cap \operatorname{ri} \operatorname{dom}(h_2) \neq \emptyset.$$

In all of the examples aboveX, this condition holds true.

9.4. Primal-dual methods for conic QP

Proposition 9.3.1 provides a means of evaluating the proximal map of QS functions via conic quadratic optimization. There are many algorithms for solving convex conic QPs, but primal-dual methods offer a particularly efficient approach that can leverage the special structure that defines the class of QS functions. A detailed discussion of the implementation of primal-dual methods for conic optimization is given by [Van10]. Here we summarize the main aspects that pertain to implementing these methods efficiently in our context.

The standard development of primal-dual methods for (9.3.1) is based on perturbing the optimality conditions, which can be stated as follows. The solution y, together with slack and dual vectors s and v, must satisfy

$$Qy - A^T v = c, \quad v \succeq_{\mathcal{K}} 0, \quad Sv = 0,$$

where the matrix S is block diagonal, and each m_i-by-m_i block S_i is either a diagonal or arrow matrix depending on the type of cone, i.e.,

$$S_i = \begin{cases} \mathbf{diag}(s_i) & \text{if } \mathcal{K}_i = \mathbf{R}_+^{m_i}, \\ \mathbf{arrow}(s_i) & \text{if } \mathcal{K}_i = \mathcal{Q}^{m_i}, \end{cases} \qquad \mathbf{arrow}(u) := \begin{pmatrix} u_0 & \bar{u}^T \\ \bar{u} & u_0 I \end{pmatrix} \quad \text{for} \quad u = (u_0, \bar{u}).$$

See [Van10] for further details. Now replace the complementarity condition $Sv = 0$ with its perturbation $Sv = \mu e$, where μ is a positive parameter and $e = (e^1, \dots, e^k)$, with each partition defined by

$$e^i = \begin{cases} (1, 1, \dots, 1) & \text{if } \mathcal{K}_i = \mathbf{R}_+^{m_i}, \\ (1, 0, \dots, 0) & \text{if } \mathcal{K}_i = Q^{m_i}. \end{cases}$$

A Newton-like method is applied to the perturbed optimality conditions, which we phrase as the root of the function

$$R_\mu : \begin{pmatrix} y \\ v \\ s \end{pmatrix} \mapsto \begin{pmatrix} r_d \\ r_p \\ r_\mu \end{pmatrix} := \begin{pmatrix} Qy - A^T v - c \\ Ay - s - b \\ Sv - \mu e \end{pmatrix}. \tag{9.4.1}$$

Each iteration of the method proceeds by systematically choosing the perturbation parameter μ (ensuring it decreases), and obtaining each search direction as the solution of the Newton system

$$\begin{pmatrix} Q & -A^T & 0 \\ A & 0 & -I \\ 0 & S & V \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta v \\ \Delta s \end{pmatrix} = - \begin{pmatrix} r_d \\ r_p \\ r_\mu \end{pmatrix}, \qquad \begin{pmatrix} y^+ \\ v^+ \\ s^+ \end{pmatrix} = \begin{pmatrix} y \\ v \\ s \end{pmatrix} + \alpha \begin{pmatrix} \Delta y \\ \Delta v \\ \Delta s \end{pmatrix}. \tag{9.4.2}$$

The steplength α is chosen to ensure that (v^+, s^+) remain in the strict interior of the cone.

One approach to solving for the Newton direction is to apply block Gaussian elimination to (9.4.2), and obtain the search direction via the following systems:

$$(Q + A^T S^{-1} V A)\Delta x = r_d + A^T S^{-1}(V r_p + r_\mu), \tag{9.4.3a}$$

$$\Delta v = S^{-1}(V r_p + r_\mu - V A \Delta x), \tag{9.4.3b}$$

$$\Delta s = V^{-1}\left(r_\mu - S\Delta v\right). \tag{9.4.3c}$$

In practice, the matrices S and V are rescaled at each iteration in order to yield search directions with favorable properties. In particular, the Nesterov-Todd rescaling redefines S and V so that $SV^{-1} = \mathbf{block}(u)$ for some vector u, where

$$\mathbf{block}(u)_i = \begin{cases} \mathbf{diag}(u_i) & \text{if } \mathcal{K}_i = \mathbf{R}^{m_i}_+, \\ (2u_iu_i^T - [u_i^T J u_i] J)^2 & \text{if } \mathcal{K}_i = \mathcal{Q}^{m_i}, \end{cases} \qquad J = \begin{pmatrix} 1 & 0 \\ 0 & -I_{(m_i-1)} \end{pmatrix}. \tag{9.4.4}$$

The cost of the overall approach is therefore determined by the cost of solving, at each iteration, linear systems with the matrix

$$\mathcal{L}(u) := Q + A^T \mathbf{block}(u)^{-1} A, \tag{9.4.5}$$

which now defines the system (9.4.3a).

9.5. Evaluating the proximal operator

We now describe how to use Proposition 9.3.1 to transform a scaled proximal operator (1.3.6) into a conic QP that can be solved by the interior algorithm described in §9.4. In particular, to evaluate $\mathbf{prox}_g^H(x)$ we solve the conic QP (9.3.1) with the definitions

$$Q := BH^{-1}B^T, \qquad c := d + Bx; \tag{9.5.1}$$

the other quantities A, b, and the cone $\mathcal{K}$, appear verbatim. Algorithm 9.1 summarizes the procedure. As we note in §9.4, the main cost of this procedure is incurred in Step 1, which requires repeatedly solving linear systems that involve the linear operator (9.4.5). Together with (9.5.1), these matrices have the form

$$\mathcal{L}(u) = BH^{-1}B^T + A^T\mathbf{block}(u)^{-1}A. \tag{9.5.2}$$

Algorithm 9.1: Evaluating $\mathbf{prox}_H^g(x)$

```
INPUT  : $x$, $H$, and QS function $g$ as defined by parameters $A$, $b$, $d$, $B$, $\mathcal{K}$
OUTPUT : $\mathbf{prox}_H^g(x)$
Step 1: Apply interior method to QP (9.3.2b) to obtain $y^*$.
Step 2: Return $H^{-1}(c - B^T y^*)$.
```

Below we offer a tour of several examples, ordered by level of simplicity, to illustrate the details involved in the application of our technique. The Sherman-Woodbury (SW) identity

$$(D + UMU^T)^{-1} = D^{-1} - D^{-1}U(M^{-1} + U^TD^{-1}U)^{-1}U^TD^{-1},$$

valid when $M^{-1} + U^TD^{-1}U$ is nonsingular, proves useful for taking advantage of certain structured matrices that arise when solving (9.5.2). Some caution is needed, however, because it is known that the SW identity can be numerically unstable [Yip86].

For our purposes, it is useful to think of the SW formula as a routine that takes the elements (D, U, M) that define a linear operator $D + UMU^T$, and returns the elements (D_1, U_1, M_1) that define the inverse operator $D_1 + U_1M_1U_1^T = (D + UMU^T)^{-1}$. We assume that D and M are nonsingular. Algorithm 9.2 summarizes the operations needed to compute the elements of the inverse operator.

Typically, D is a structured operator that admits a fast algorithm for solving linear systems with any right-hand side, and U and M are stored explicitly as dense matrices. Step 1 computes a new operator D_1 that simply interchanges the multiplication and inversion operations of D. Step 2 applies the operator D_1 to every column of U (typically a tall matrix with few columns). Step 3 requires inverting a small matrix.

Algorithm 9.2: Inverse via the Sherman-Woodbury identity

```
function SWinv($D, U, M$)
1    $D_1 \leftarrow D^{-1}$
2    $U_1 \leftarrow D_1 U$
3    $M_1 \leftarrow (M^{-1} + U^T U_1)^{-1}$
4    return $D_1$, $U_1$, $M_1$
end
```

Example 9.5.1 (1-norm regularizer; cf. Example 9.1.1)**.** Example 9.1.1 gives the QS representation for $g(x) = \|x\|_1$, and the required expressions for A, B, and $\mathcal{K}$. Because $\mathcal{K}$ is the nonnegative orthant, $\mathbf{block}(u) = \mathbf{diag}(u)$; cf. (9.4.4). With the definitions of A and B, the linear operator $\mathcal{L}$ in (9.5.2) simplifies to

$$\mathcal{L}(u) = H^{-1} + A^T \mathbf{diag}(u) A = H^{-1} + \Sigma,$$

where Σ is a positive-definite diagonal matrix that depends on u. If it happens that the preconditioner H has a special structure such that $H + \Sigma^{-1}$ is easily invertible, it may be convenient to apply the SW identity to obtain equivalent formulas for the inverse

$$\mathcal{L}(u)^{-1} = (H^{-1} + \Sigma)^{-1} = H - H(H + \Sigma^{-1})^{-1} H.$$

Banded, chordal, and diagonal-plus-low-rank matrices are examples of specially structured matrices that make one of these formulas for $\mathcal{L}^{-1}$ efficient. They yield the efficiency because subtracting the diagonal matrix Σ preserves the structure of either H or H^{-1}.

In the important special case where $H = \mathbf{diag}(h)$ is diagonal, each component i of the proximal operator for the 1-norm can be obtained directly via the formula

$$[\mathbf{prox}_g^H(x)]_i = \mathbf{sign}(x_i) \cdot \max\{|x_i - 1|/h_{ii},\, 0\},$$

where h_{ii} are the diagonal elements of H. This corresponds to the well-known soft-thresholding operator. No simple formula exists, however, for more general matrices.

Example 9.5.2 (Graph-based 1-norm)**.** Consider the graph-based 1-norm function from Example 9.2.2 induced by a graph $\mathcal{G}$ with adjacency matrix N. Substitute the definitions of A and B from (9.2.2) into the formula for $\mathcal{L}$ and simplify to obtain

$$\mathcal{L}(u) = NH^{-1}N^T + A^T\mathbf{diag}(u)A = NH^{-1}N^T + \Sigma,$$

where $\Sigma := A^T\mathbf{diag}(u)A$ is a positive-definite diagonal matrix. (As with Example 9.5.1, $\mathcal{K}$ is the positive orthant, and thus $\mathbf{block}(u) = \mathbf{diag}(u)$.) Linear systems of the form $\mathcal{L}(u)p = q$ then can be solved with the following sequence of operations outlined in Algorithm 9.3, in which we assume that $H = \Lambda + UMU^T$, where Λ is diagonal.

Algorithm 9.3: Solving the system $\mathcal{L}(u)p = q$ for the graph-based 1-norm.

```
1 $(\Lambda_1, U_1, M_1) \leftarrow \texttt{SWinv}(\Lambda, U, M)$        [ $H^{-1} \equiv \Lambda_1 + U_1M_1U_1^T$ ]
2 $\Sigma_1 \leftarrow N\Lambda_1N^T + \Sigma$
3 $(\Sigma_2, U_2, M_2) \leftarrow \texttt{SWinv}(\Sigma_1, NU_1, M_1)$   [ $\mathcal{L}(u)^{-1} \equiv \Sigma_2 + U_2M_2U_2^T$ ]
4 $p \leftarrow \Sigma_2q + U_2M_2U_2^Tq$                       [solve $\mathcal{L}(u)p = q$]
5 return $p$
```

Observe from the definition of H and the definition of Σ_1 in Step 2 that

$$\mathcal{L}(u) = \Sigma_1 + NU_1M_1U_1^TN^T,$$

and then Step 3 computes the quantities that define the inverse of $\mathcal{L}$. The bulk of the work in the above algorithm happens in Step 3, where $\Sigma_2 \equiv \Sigma_1^{-1}$ is applied to each column of NU_1 (see Step 2 of the `SWinv` function), and in Step 4, where Σ_2 is applied to q. Below we give two special cases where it is possible to take advantage of the structure of N and H in order to apply Σ_2 efficiently to a vector.

1-dimensional total variation: Suppose that the graph $\mathcal{G}$ is a path. Then the $(n-1)\times n$ adjacency matrix is given by

$$N = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}.$$

The matrix $\Sigma_1 := N\Lambda^{-1}N^T + \Sigma$ (see Step 2 of the above algorithm) is tridiagonal, and hence equations of the form $\Sigma_1 q = p$ can be solved efficiently using standard techniques, e.g., [GL89, Algorithm 4.3.6].

Chordal graphs: If the graph $\mathcal{G}$ is chordal, than the matrix N^TDN is also chordal when D is diagonal. This implies that it can be factored in time linear with the number of edges of the graph [ADV10]. We can use this fact to apply $\Sigma_2 \equiv \Sigma_1^{-1}$ efficiently, as follows: let $(\Sigma_3, U_3, M_3) = \texttt{SWinv}(\Sigma, N, \Lambda_1)$, which implies

$$\Sigma_2 := \Sigma_3 + U_3M_3U_3^T, \quad \text{where} \quad \Sigma_3 := \Sigma^{-1}, \quad M_3 := (N^T\Sigma^{-1}N + \Lambda_1)^{-1}.$$

Because $N^T\Sigma^{-1}N$ is chordal, so is M_3, and any methods efficient for solving with chordal matrices can be used when applying Σ_2.

Example 9.5.3 (1-norm constraint; cf. Example 9.1.5)**.** Example 9.1.5 gives the QS representation for the indicator function on the 1-norm ball. Because the constraints on y in (9.1.4) involve only bound constraints, $\mathbf{diag}(u) = \mathbf{diag}(u)$. With the definitions of A and B from Example 9.1.5, the linear operator $\mathcal{L}$ has the form

$$\mathcal{L}(u) = \begin{pmatrix} 0 \\ I_n \end{pmatrix} H^{-1} \begin{pmatrix} 0 & I_n \end{pmatrix} + \begin{pmatrix} \mathbf{1}_n^T & \mathbf{1}_n^T \\ -I_n & I_n \end{pmatrix} \begin{pmatrix} \mathbf{diag}(u_1) & \\ & \mathbf{diag}(u_2) \end{pmatrix} \begin{pmatrix} \mathbf{1}_n & -I_n \\ \mathbf{1}_n & I_n \end{pmatrix},$$

where $u = (u_1, u_2)$. Thus, $\mathcal{L}$ simplifies to

$$\mathcal{L}(u) = \begin{pmatrix} \mathbf{1}_n^T u & (u^-)^T \\ u^- & H^{-1} + \Sigma \end{pmatrix} \qquad \text{where} \qquad \Sigma := \mathbf{diag}(u^+), \qquad \begin{aligned} u^+ &:= u_1 + u_2, \\ u^- &:= -u_1 + u_2. \end{aligned}$$

Systems that involve $\mathcal{L}$ can be solved by pivoting on the block $(H^{-1} + \Sigma)$. The cases where this approach is efficient are exactly those that are efficient in the case of Example 9.5.1.

Example 9.5.4 (2-norm; cf. Example 9.1.2)**.** Example 9.1.2 gives the QS representation for the 2-norm function. Because $\mathcal{K} = Q^n$, then $\mathbf{block}(u) = (2uu^T - [u^T Ju]J)^2$, where $u = (u_0, \bar{u})$ and J is specified in (9.4.4). With the expressions for A and B from Example 9.1.2, the linear operator $\mathcal{L}$ reduces to

$$\mathcal{L}(u) = H^{-1} + \alpha I_n + vv^T, \quad \text{with} \quad \alpha = (u^T Ju)^2,\ v = \sqrt{8u_0} \cdot \bar{u}. \tag{9.5.3}$$

This amounts to a perturbation of H^{-1} by a multiple of the identity, followed by a rank-1 update. Therefore, systems that involve $\mathcal{L}$ can be solved at the cost of solving systems with $H + \alpha I_n$ (for some scalar α).

Of course, the proximal map of the 2-norm is easily computed by other means; our purpose here is to use this as a building block for more useful penalties, such as Example 9.2.1, which involves the sum-of-norms function shown in (3.1.16). Suppose that the p partitions do not overlap, and have size n_i for $i = 1, \ldots, p$. The operator $\mathcal{L}$ in (9.5.3) generalizes to

$$\mathcal{L}(u) = H^{-1} + \underbrace{\begin{pmatrix} \alpha_1 I_{n_1} + v^1(v^1)^T & & \\ & \ddots & \\ & & \alpha_p I_{n_p} + v^p(v^p)^T \end{pmatrix}}_{W}, \quad \begin{aligned} u^i &= (u_0^i, \bar{u}^i) \\ \alpha_i &= (u^{iT} J u^i)^2 \\ v^i &= \sqrt{8u_0^i} \cdot \bar{u}^i, \end{aligned}$$

where each vector u_i has size $n_i + 1$.

When p is large, we can treat each diagonal block of W as an individual (small) diagonal-plus-rank-1 matrix. If H^{-1} is diagonal-plus-low-rank, for example, the diagonal part of H^{-1} can be subsumed into W. In that case, each diagonal block in W remains diagonal-plus-rank-1, which can be inverted in parallel by handling each block individually. Subsequently, the inverse of $\mathcal{L}$ can be obtained by a second correction.

Another approach, when p is small, is to consider W as a diagonal-plus-rank-p matrix:

$$W = \begin{pmatrix} \alpha_1 I_{n_1} & & \\ & \ddots & \\ & & \alpha_p I_{n_p} \end{pmatrix} + \begin{pmatrix} v^1 & & \\ & \ddots & \\ & & v^p \end{pmatrix} \begin{pmatrix} v^1 & & \\ & \ddots & \\ & & v^p \end{pmatrix}^T .$$

This representation is convenient: systems involving $\mathcal{L}$ can be solved efficiently in a manner identical to that of Example 9.5.1 because W is a diagonal-plus-low-rank matrix.

Example 9.5.5 (separable QS functions)**.** Suppose that g is separable, i.e.,

$$g(x) = \gamma(x_1) + \cdots + \gamma(x_n),$$

where $\gamma : \mathbf{R} \to \mathbf{R}$ is a QS function with parameters $(A_\gamma, b_\gamma, B_\gamma, d_\gamma, \mathbf{R}^{np}_+)$, and p is an integer parameter that depends on γ. The parameters A and B for g follow from the concatenation rule (9.2.1), and $A = (I_n \otimes A_\gamma)$ and $B = (I_n \otimes B_\gamma)$. Thus, the linear operator $\mathcal{L}$ is given by

$$\mathcal{L}(u) = (I_n \otimes B_\gamma) H^{-1} (I_n \otimes B_\gamma)^T + (I_n \otimes A_\gamma)^T \mathbf{diag}(u)(I_n \otimes A_\gamma).$$

Apply the SW identity to obtain

$$\mathcal{L}(u)^{-1} = \Lambda^{-1} - \Lambda^{-1}(I_n \otimes B_\gamma)(H + \Sigma)^{-1}(I_n \otimes B_\gamma)^T \Lambda^{-1},$$

where $\Lambda = \mathbf{diag}(\Lambda_1, \ldots, \Lambda_n)$,

$$\Lambda_i = A_\gamma^T \mathbf{diag}(u^i) A_\gamma, \quad \text{and} \quad \Sigma = \mathbf{diag}(B_\gamma^T \Lambda_1^{-1} B_\gamma, \ldots, B_\gamma^T \Lambda_n^{-1} B_\gamma).$$

Because the function γ takes a scalar input, B_γ is a vector. Hence Σ is a diagonal matrix. Note too that Λ is a block diagonal matrix with n blocks each of size p. We can then solve the system $\mathcal{L}(u)p = q$ with the following steps:

```
1 $q_1 \leftarrow (I_n \otimes B_\gamma)^T \Lambda^{-1} q$
2 $q_2 \leftarrow (H + \Sigma)^{-1} q_1$
3 $q_3 \leftarrow \Lambda^{-1} q_2 - \Lambda^{-1}(I_n \otimes B_\gamma) q_2$
```

The cost of solving systems with the operator $\mathcal{L}$ is dominated by solves with the block diagonal matrix Λ (Steps 1 and 3) and $H + \Sigma$ (Step 2). The cost of the latter linear solve is explored in Example 9.5.1.

9.6. A proximal quasi-Newton method

We now turn to the proximal-gradient method discussed in Section 1.3.2.1. Our primary goal is to demonstrate the feasibility of the interior approach for evaluating proximal operators of QS functions. A secondary goal is to illustrate how this technique leads to an efficient extension of the quasi-Newton method for nonsmooth problems of practical interest.

We follow [ST16] and implement a limited-memory BFGS (L-BFGS) variant of the proximal-gradient method that has no linesearch and that approximately evaluates the proximal operator. Scheinberg and Tang establish a sublinear rate of convergence for this method when the Hessian approximations are suitably modified by adding a scaled identity matrix, and when the scaled proximal maps are evaluated with increasing accuracy. In their proposal, the accuracy of the proximal evaluation is based on bounding the value of the approximation in the function value of (2.2.1). We depart from this criterion, however, and instead use the residual (9.4.1) obtained by the interior solver to determine the required accuracy. In particular, we require that the optimality criterion of the interior algorithm used to evaluate the operator is a small multiplicative constant κ of the current optimality of the outer proximal-gradient iterate, i.e.,

$$\|R_\mu(y, v, s)\| \le \kappa \|x_k - \mathbf{prox}_g(x_k - \nabla f(x_k))\|.$$

This heuristic is reminiscent of the accuracy required of the linear solves used by an inexact Newton method for root finding [DES82]. Note that the proximal map $\mathbf{prox}_g \equiv \mathbf{prox}_g^I$ used above is unscaled, which in many cases can be easily computed when g is separable.

9.6.1. Limited-memory BFGS updates.

Each interior iteration for evaluating the proximal operator depends on solving linear systems with $\mathcal{L}$ in (9.5.2). In all of the experiments presented below, each interior iteration has a cost that is linear in the number of variables n.

9.7. Numerical experiments

We have implemented the proximal quasi-Newton method as a Julia package [BEKS14], called `QSip` designed for problems of the form (11.2.1a), where f is smooth and g is a QS function. The code is available at the URL

`https://github.com/MPF-Optimization-Laboratory/QSip.jl`

A primal-dual interior method, based on ideas from the CVXOPT software package [ADV10], is used for Algorithm 9.1. We consider below several examples. The first three examples apply the `QSip` solver to minimize benchmark least-squares problems with different nonsmooth regularizers that are QS representable; the last example applies the solver to a sparse logistic-regression problem on a standard data set.

9.7.1. Timing the proximal operator. The examples that we explored in §9.5 have a favorable structure that allows each interior iteration for evaluating the proximal map $\mathbf{prox}_g^H(x)$ to scale linearly with problem size. In this section we verify this behavior empirically for problems with the structure

$$H = I + UU^T, \qquad g(x) = \|x\|_1, \qquad U \in \mathbf{R}^{n\times k} \tag{9.7.1}$$

for different values of k and n. This choice of diagonal-plus-low-rank matrices is designed to mimic the structure of matrices that appear in L-BFGS. Here U and x are chosen with random normal entries. As described in Example 9.1.1, the system $\mathcal{L}(u)$ is inverted in linear time using the SW identity.

We evaluate the proximal map on 100 random instances for each combination of k and n, and plot in Figure 1 the average time needed to reach an accuracy of 10^{-7}, as measured by the optimality conditions in the interior algorithm. Because in practice the number of iterations of the interior method is almost independent of the size of the problem, the time taken to compute the proximal map is a predictable, linear function of the size of the problem.

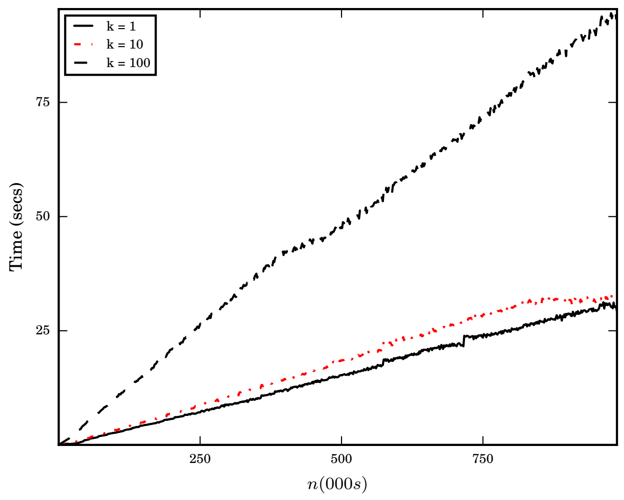


FIGURE 9.7.1. Time taken to compute $\mathbf{prox}_g^H(x)$ versus n, for $k = 1, 10, 100$; see (9.7.1).

9.7.2. Synthetic least-square problems. The next set of examples all involve the least-squares objective

$$f(x) = \tfrac{1}{2}\|Ax - b\|_2^2. \tag{9.7.2}$$

Two different procedures are used to construct matrices A, as described in the following sections. In all cases, we follow the testing approach described by [Lor13] for constructing a test problem with a known solution: fix a vector $x^\star$ and choose $b = Ax^\star - A^{-T}v$, where $v \in \partial g(x^\star)$. Note that

$$\partial(f + g)(x^\star) = A^T(Ax^\star - [Ax^\star - A^{-T}v]) + \partial g(x^\star) = \partial g(x^\star) - v.$$

Because $v \in \partial g(x^\star)$, the above implies that $0 \in \partial(f + g)(x^\star)$, and hence $x^\star$ minimizes the objective $f + g$. In the next three sections, we apply `QSip` in turn to problems with g equal to the 1-norm, the group LASSO (i.e., sum of 2-norm functions), and total variation.

9.7.2.1. *One-norm regularization.* In this experiment we choose $g = \|\cdot\|_1$, which gives the 1-norm regularized least-squares problem, often used in applications of sparse optimization. Following the details in Example 9.1.1, the system $\mathcal{L}(u)$ is a diagonal-plus-low-rank matrix, which we invert using the SW identity.

The matrix A in (9.7.2) is a 2000-by-2000 lower triangular matrix with all nonzero entries equal to 1. The bandwidth p of A is adjustable, and determines its coherence

$$\text{coherence}(A) = \max_{i \neq j} \frac{a_i^T a_j}{\|a_i\| \|a_j\|} = \sqrt{\frac{p-1}{p}},$$

where a_i is the ith column. As observed by [Lor13], the difficulty of 1-norm regularized least-squares problems are strongly influenced by the coherence. Our experiments use matrices A with bandwidth $p = 500, 1000, 2000$.

Figure 9.7.2 shows the results of applying the `QSip` solver with a memories $k = 1, 10$, labeled "QSIP mem $= k$". We also consider comparisons against two competitive proximal-based methods. The first is a proximal-gradient algorithm that uses the Barzilai-Borwein steplength [BB88, WNF09]. This is our own implementation of the method, and is labeled "Barzilai-Borwein" in the figures. The second is the proximal quasi-Newton method implemented by [BF12], which is based on a symmetric-rank-1 Hessian approximation; this code is labeled "PG-SR1". The `QSip` solver with memory of 10 outperforms the other solvers. The quasi-Newton approximation benefits problems with high coherence (p large) more than problems with low coherence (p small). In all cases, the experiments reveal that the additional cost involved in evaluating a proximal operator (via an interior method) is balanced by the overall cost of the algorithm, both in terms of iterations (i.e., matrix-vector products with A) and time.

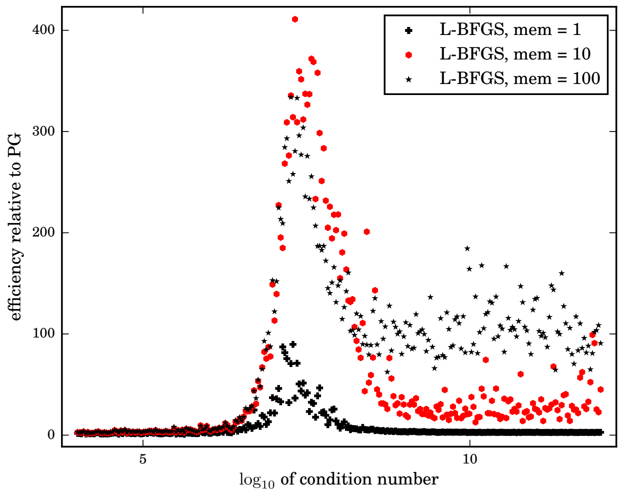


FIGURE 9.7.3. Performance of the proximal quasi-Newton method relative to proximal gradient for problems of varying condition number.

9.7.2.2. *The effect of conditioning.* It is well known that the proximal-gradient method converges slowly for ill conditioned problems. The proximal L-BFGS method may help to improve convergence in such situations. We investigate the observed convergence rate of the proximal L-BFGS approach on a family of least-squares problems with 1-norm regularization with varying degrees of ill conditioning. For these experiments, we take A in (9.7.2) as the 2000-by-2000 matrix

$$A = \alpha_L \begin{pmatrix} T & 0 \\ 0 & 0 \end{pmatrix} + \alpha_\mu I,$$

where T is a 1000-by-1000 tridiagonal matrix with constant diagonal entries equal to 2, and constant sub- and super-diagonal entries equal to -1. The parameter α_L/α_μ controls the conditioning of A, and hence the conditioning of the Hessian $A^T A$ of f.

We run L-BFGS with 4 different memories ("mem"): 0 (i.e., proximal gradient with a Barzilai-Borwein steplength), 1, 10, and 100. We terminate the algorithm either when the error drops beneath 10^{-8}, or the method reaches 10^3 iterations. Our method of measuring the observed convergence (OC) computes the line of best fit to the log of optimality versus k, which results in the quantity

$$\text{Observed Convergence} := \frac{\sum_{k=0}^{N} k \cdot \log \|x_k - x_*\|}{\sum_{k=0}^{N} \log \|x_k - x_*\|},$$

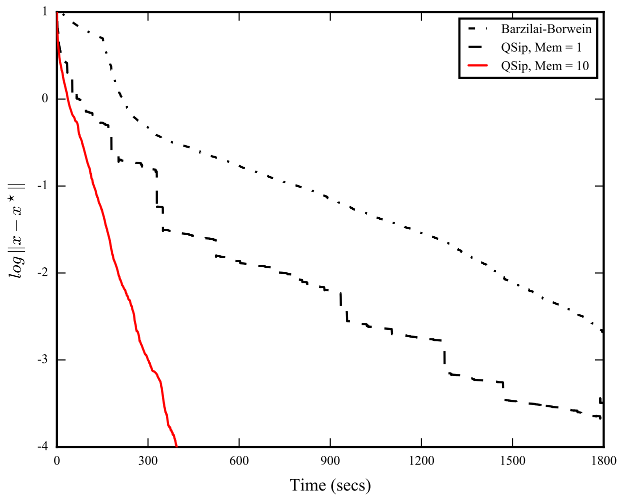
where N is the total number of iterations.

The plot in Figure 9.7.3 shows the ratio of the OC for L-BFGS relative to the observed convergence of proximal gradient (PG). This quantity can be interpreted the amount of work that a single quasi-Newton step performs relative to the number of PG iterations. The plot reveals that the quasi-Newton method is faster at all condition numbers, but is especially effective for problems with moderate conditioning. Also, using a higher quasi-Newton memory almost always lowers the number of iterations. This benefit is most pronounced when the problem conditioning is poor.

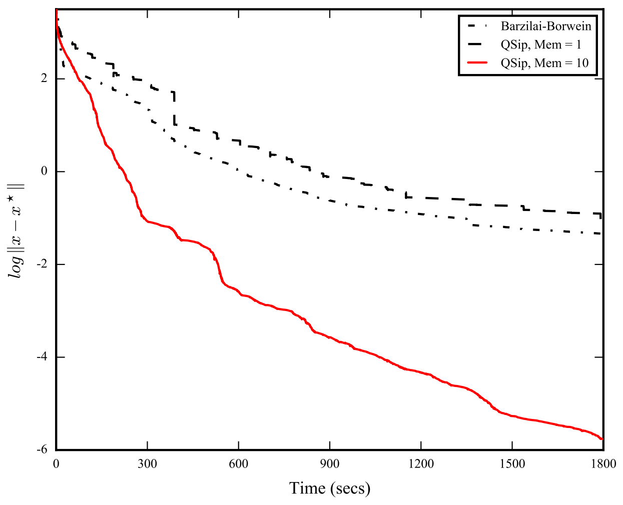
Together with §9.7.1, this section gives a broad picture of the trade-off between the proximal quasi-Newton and proximal gradient methods. The time required for each proximal gradient iteration is dominated by the cost of the gradient computation because the evaluation of the unscaled proximal operator is often trivial. On the other hand, the proximal quasi-Newton iteration additionally requires evaluating the scaled proximal operator. Therefore, the proximal quasi-Newton method is most appropriate when this cost is small relative to the gradient evaluation.

9.7.2.3. *Group LASSO.* Our second experiment is based on the sum-of-norms regularizer described in Examples 9.2.1 and 9.5.4. In this experiment, the n-vector (with $n = 2000$) is partitioned into $p = 5$ disjoint blocks of equal size. The matrix A is fully lower triangular.

Figure 9.7.4a clearly shows that the `QSip` solver outperforms the PG method with the Barzilai-Borwein step size. Although we required `QSip` to exit with a solution estimate accurate within 6 digits (i.e., $\log \|x - x^*\| \leq 10^{-6}$), the interior solver failed to achieve the requested accuracy because of numerical instability with the SW formula used for solving the Newton system. This raises the question of how to use efficient alternatives to the SW update that are numerically stable and can still leverage the structure of the problem.



(a) Performance of solvers applied to a group-Lasso problem.



(b) Performance of the `QSip` solver applied to a 1-dimensional total-variation problem.

FIGURE 9.7.4. The horizontal axis measures elapsed time; the vertical axis measures distance to the solution.

9.7.2.4. *1-dimensional total variation.* Our third experiment sets

$$g(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|,$$

which is the anisotropic total-variation regularizer described in Examples 9.2.2 and 9.5.2. The matrix A is fully lower triangular. Figure 9.7.4b compares the convergence behavior of `QSip` with the Barzilai-Borwein proximal solver. The Python package `prox-tv` [BS11, BS14] was used for the evaluation of the (unscaled) proximal operator, needed by the Barzilai-Borwein solver. The `QSip` solver, with memories of 1 and 10, outperformed the Barzilai-Borwein solver.

9.7.3. Sparse logistic regression. This next experiment tests `QSip` on the sparse logistic-regression problem

$$\underset{x}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N} \log(1+\exp[a_i^T x]) + \lambda\|x\|_1,$$

where N is the number of observations. The *Gisette* [GGBHD04] and *Epsilon* [Pas16] datasets, standard benchmarks from the UCI Machine Learning Repository [Lic13], are used for the feature vectors a_i. *Gisette* has $5K$ parameters and $13.5K$ observations; *Epsilon* has $2K$ parameters with $400K$ observations. These datasets were chosen for their large size and modest number of parameters. In all of these experiments, $\lambda = 0.01$.

Figure 9.7.5 compares `QSip` to the Barzilai-Borwein solver, and to newGLMNet [KKL$^+$07], a state-of-the-art solver for sparse logistic regression. (Other possible comparisons include the implementation of [ST16], which we do not include because of difficulty compiling that code.) Because we do not know a priori the solution for this problem, the vertical axis measures the log of the optimality residual $\|x_k - \mathbf{prox}_g(x_k - \nabla f(x_k))\|_\infty$ of the current iterate. (The norm of this residual necessarily vanishes at the solution.) On the *Gisette* dataset, Barzilai-Borwein and newGLMNNet are significantly faster than the proximal quasi-Newton implementation. On the *Epsilon* dataset, however, the quasi-Newton is faster at all levels of accuracy.

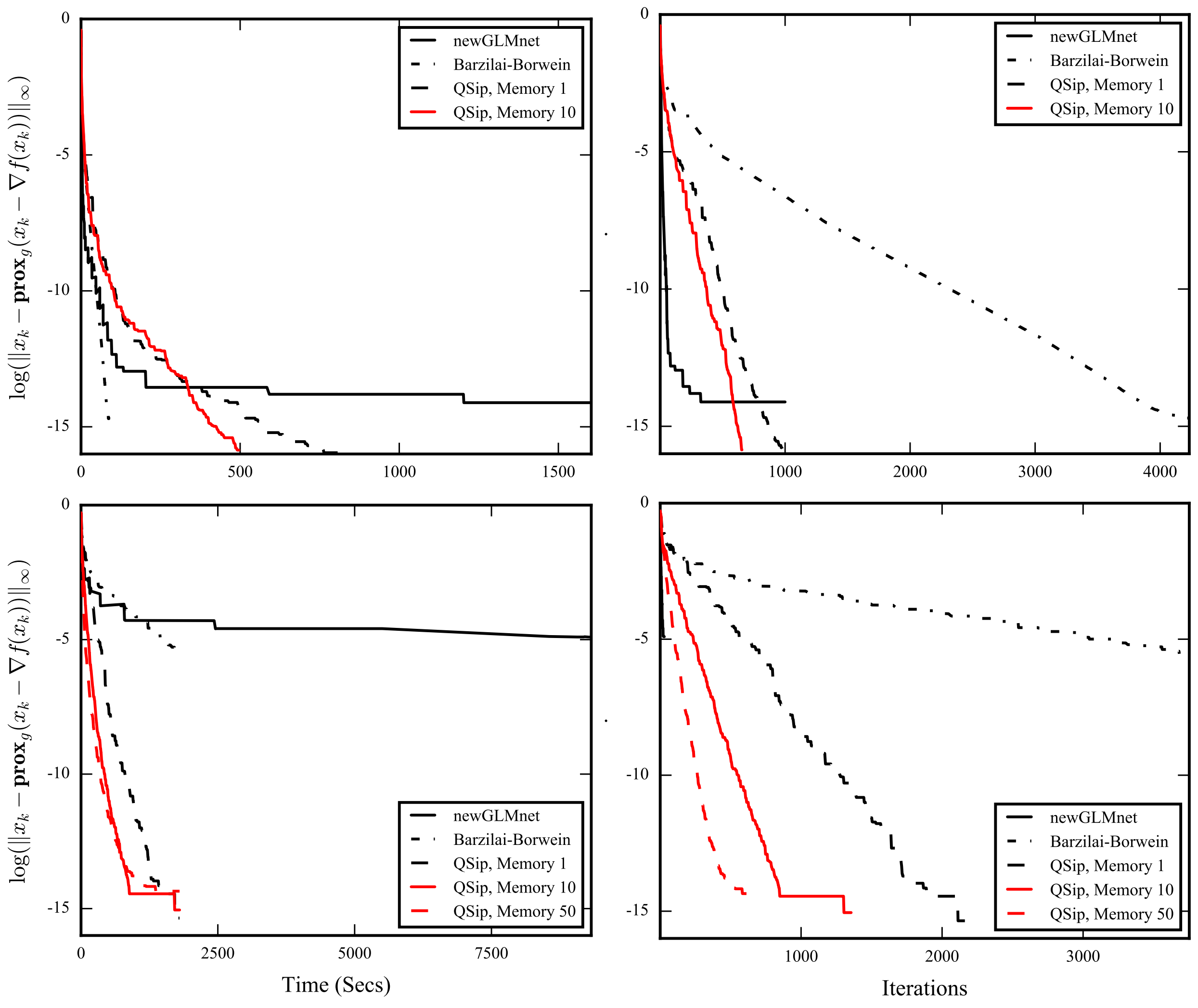


FIGURE 9.7.5. Performance of solvers on a sparse logistic-regression problem. Top row: *Gisette* dataset; bottom row: *Epsilon* dataset. The left and right columns, respectively, track the optimality of the current solution estimate versus elapsed time and iteration number.

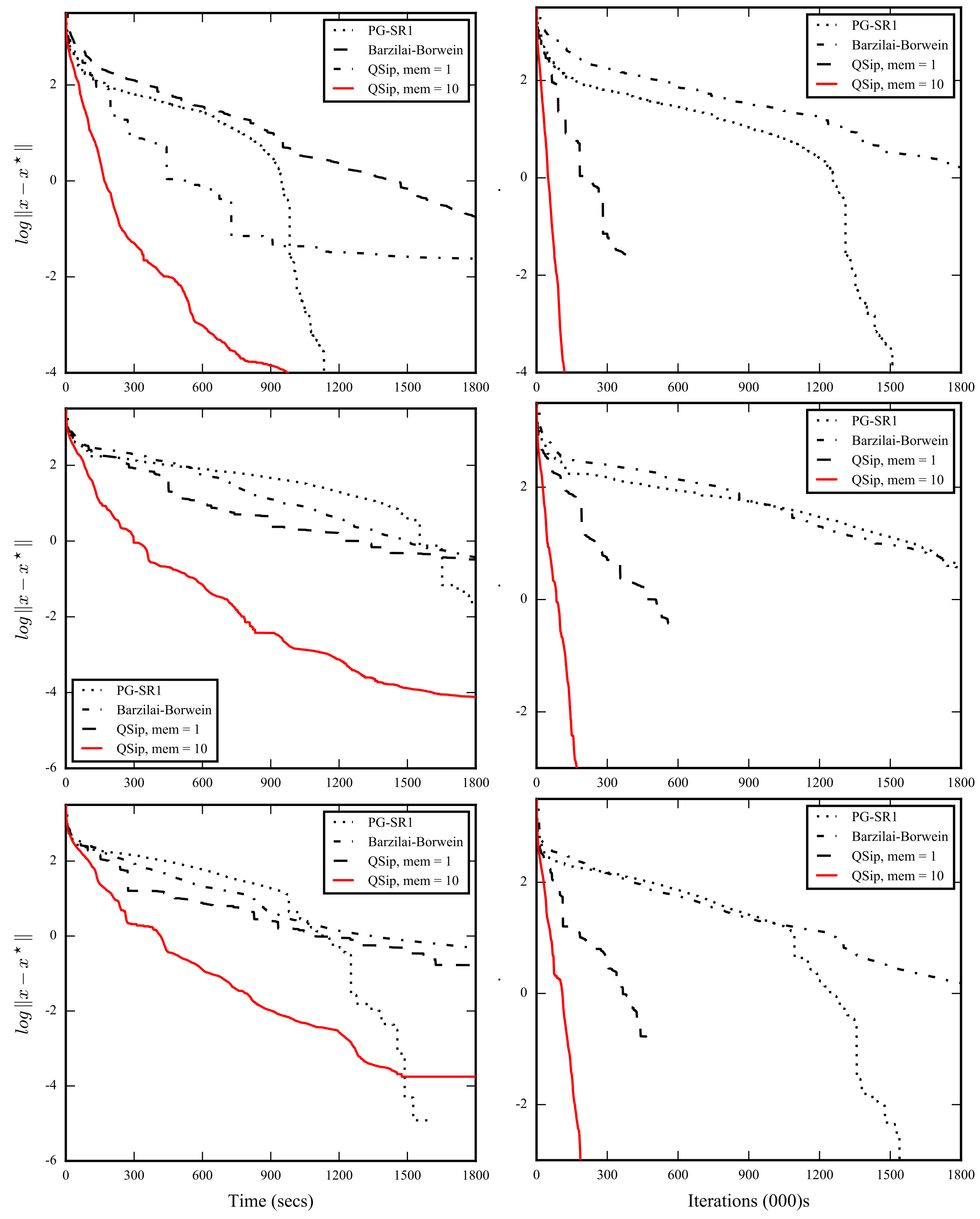


FIGURE 9.7.2. Top row: $p = 2000$ middle row: $p = 1000$; bottom row: $p = 500$.

CHAPTER 10

Stochastic oracles

We return to the discussion in the introduction, Section 1.3, of functions of the form

$$f(x) := \mathbf{E}\, h(x,\xi) = \int h(x,\xi)\, d\xi, \tag{10.0.1}$$

where $h(\cdot,\xi)$ is smooth and convex for all ξ. We define a stochastic first-order oracle as one which gives, for some tolerance ϵ, an unbiased estimator of $\nabla f(x)$ which increases in accuracy as ϵ approaches 0.

Definition 10.0.1 (Stochastic First-Order Oracle)**.** f is equipped with an inexact oracle of type A, B and C if we can find for any pair (x,ϵ) we can find a $g = \nabla f(x) + e(x)$ where $e(x)$ is a random variable which depends on x, $\mathbf{E}[e(x)] = 0$, and

A. [Variance] $\mathbf{E}\|e(x)\|^2 \le \epsilon$;

B. [High Probability] $\Pr\left(e(x)_i \ge \delta\right) \le \exp\left(-\delta^2/\epsilon\right)$ for all i, x

C. [Deterministic] $\|e(x)\|^2 \le \epsilon$.

We refer to these as A-**sto-oracle**$_f$, B-**sto-oracle**$_f$ and C-**sto-oracle**$_f$ respectively.

These oracles are ordered in increasing strength: if we have an oracle of type C-**sto-oracle**, we have an oracle of type B-**sto-oracle** holds by Serfling's inequality (Theorem B.2.2). Furthermore if we have an oracle of type B-**sto-oracle**, then we have an oracle of type A-**sto-oracle** because the exponential bound implies a bound on the second moment, i.e.,

$$\mathbf{E}\big[[e(x)]_i^2\big] = \int_0^\infty \Pr([e(x)]_i^2 \ge \delta)\, d\delta \le \int_0^\infty \exp\left(-\delta^2/\epsilon\right) d\delta < \infty.$$

10.1. Population Assumptions

The degree to which equation (10.0.1) can be estimated depends on the amount of variation present within $h(x, \cdot)$. One one extreme, if $h(x, \cdot) = h_0(x)$, a single sample is sufficient to obtain the true gradient. On the other hand, if $h(x, \cdot)$ were not integrable in ξ, the variation would simply be too large for $\nabla f(x)$ to be approximable. Real applications of stochastic gradient descent fall somewhere between these two extremes, and the following hypothesis attempts to quantify the variation within $h(x, \cdot)$, in increasing order of strength.

Hypothesis 10.1.1 (Uniform bounds)**.** For all x, ξ one of the following hypothesis hold

A. [Variance] $\frac{1}{m}\sum_{i=0}^{m} \mathbf{E}\|\nabla_x h(x,\xi_i) - \mathbf{E}[\nabla_x h(x,\xi_i)]\|^2 \leq B_A$;

B. [Exponential Tail] $\sup_x \{\max_\xi [\nabla_x h(x,\xi)]_i - \min_\xi [\nabla_x h(x,\xi)]_i\} < B_B$,

C. [Deterministic] $\|\nabla_x h(x,\xi)\|^2 \leq B_C$.

10.2. Sampling With Replacement

Since the gradient is a linear operator, under very weak conditions we can move the gradient inside the expectation, $\nabla_x \mathbf{E} h(x,\xi) = \mathbf{E}\nabla_x h(x,\xi)$. And hence if we could obtain a uniform sample from ξ_i, we can construct an estimate of the gradient sampling m gradients at these instantiations:

$$g = \mathbf{oracle}_f(x,\epsilon), \quad g = \sum_{i=0}^{m} \nabla_x h(x,\xi_i), \quad e(x) = \frac{1}{m}\sum_{i=0}^{m} \nabla_x h(x,\xi_i) - \mathbf{E}[\nabla_x h(x,\xi_i)]$$

Our error term $e(x)$ for this estimator is sampling error, which clearly has expectation 0.

(A) Variance Stochastic Oracle: The weakest of the three oracles only requires Hypothesis 10.1.1(A), a bound on the variance of the problem. Taking expectations of $\|e(x)\|$,

$$\mathbf{E}\|e(x)\|^2 = \frac{1}{m^2}\sum_{i=0}^{m} \mathbf{E}\|\nabla_x h(x,\xi_i) - \mathbf{E}[\nabla_x h(x,\xi_i)]\|^2 \leq \frac{B_A}{m}$$

Thus, to achieve an error of ϵ, one needs at least

$$m = \left\lceil \frac{B_A}{\epsilon} \right\rceil \quad \text{samples.}$$

(B) High Probability Stochastic Oracle: The high probability bounds stem from Hoeffling's inequality, applied with random variable $X_i = \nabla_x f(x, \xi_i)_i$

$$\Pr\left(e(x)_i \geq \delta\right) = \Pr\left(\nabla_x h(x, \xi_i)_i - \mathbf{E}[\nabla_x h(x, \xi_i)_i] \geq \delta\right) \leq \exp\left(\frac{-2m\delta^2}{B_B^2}\right)$$

which shows that if a random variable is bounded, its tails are not heavy. Therefore, if Hypothesis 10.1.1(A) holds, then we need

$$m = \left\lceil \frac{B_B^2}{2\epsilon} \right\rceil \qquad \text{samples}$$

to get a high probability oracle.

(C) Deterministic Stochastic Oracle: We cannot construct a controllable deterministic oracle. No matter how many samples we take, there is a small but nonzero chance the error will not drop - for example if the same sample repeated every time.

10.3. Sampling Without Replacement

When ξ takes on a finite number of values with uniform probability, then f is equivalent to the familiar case of sums of functions

$$h(x, \xi) = \frac{1}{M} \sum_{i=1}^{M} h_i(x), \tag{10.3.1}$$

When the sum is finite, we can improve over independent sampling - we can take samples without replacement. This reduces the number of samples we need to control ϵ, because as we approach M, our population size, our error goes to 0.

(A) Variance Stochastic Oracle: If Hypothesis 10.1.1 (A) holds

$$\mathbf{E}\|e(x)\|^2 = \left(\frac{M-m}{M}\right) \frac{\frac{1}{m}\sum_{i=0}^{m} \mathbf{E}\|\nabla h_i(x) - \mathbf{E}[\nabla h_i(x)]\|^2}{m} \leq \left(\frac{M-m}{M}\right) \frac{B_A}{m}$$

Notice this differs from the previous bound by a factor of $1 - m/M$. This term, known as the finite population correction, approaches 0 as m approaches M, shrinking the variance of the estimator dramatically in that regime. Therefore there are

$$m = \left\lceil \frac{B_A(B_A+1)}{B_A+2M\epsilon} \right\rceil$$

samples needed to get this oracle.

(B) High Probability Stochastic Oracle: The high probability bounds require a variation of Hoeffding's inequality, Serfling's inequality, Theorem B.2.2, which show

$$\Pr\left(e(x)_i \geq \delta\right) = \Pr\left(\nabla_x h(x,\xi_i)_i - \mathbf{E}[\nabla_x h(x,\xi_i)_i] \geq \delta\right) \leq \exp\left(\frac{-2m\delta^2}{B_B^2}\right)$$

Where the final step requires the assumption (C). Therefore we require

$$m = \left\lceil \frac{B_B(B_B+1)}{B_B+2M\epsilon} \right\rceil \qquad \text{samples.}$$

(C) Deterministic Stochastic Oracle: If we assume (C) then the sampling error can be bounded by, as shown in [FS12],

$$\begin{aligned} \|e(x)\|^2 &= \left\| \frac{M-m}{Mm} \sum_{i\in S} \nabla h_i(x) - \frac{1}{M} \sum \nabla h_i(x) \right\|^2 \\ &\leq \left(\frac{M-m}{Mm} \left\| \sum_{i\in S} \nabla h_i(x) \right\| + \frac{1}{M} \left\| \sum_{i=1}^{M} \nabla h_i(x_k) \right\| \right)^2 \leq 4\left(1-\frac{m}{M}\right)^2 B_C \end{aligned} \tag{10.3.2}$$

Therefore we require

$$m = \left\lceil B_C\left(1-\sqrt{\frac{\epsilon}{4}}\right) \right\rceil \qquad \text{samples}$$

to achieve a deterministic oracle.

CHAPTER 11

Stochastic gradient descent

Here we study our proximal gradient iteration when the gradient is replaced by the approximation,

$$x_{k+1} = \mathbf{prox}_{g/\alpha_k}(x_k - \alpha_k^{-1} s_k), \qquad \text{where} \qquad s_k = A/B/C\text{-}\mathbf{sto\text{-}oracle}(x_k, \epsilon_k), \tag{11.0.1}$$

for some error schedule ϵ_k. Our aim in this chapter is to bound the probability that the rate of convergence of the stochastic method deviates from linear-convergence rate, i.e., we provide tail bounds on

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \qquad \text{where} \qquad \pi_k = f(x_k) + g(x_k) - \min_x \{f(x) + g(x)\}. \tag{11.0.2}$$

It is straightforward to recast these results to obtain bounds on $\Pr(\pi_k \geq \epsilon)$, i.e. confidence levels for the solution x_k, (1.3.9). In Section 11.2 we describe bounds that depend on the errors generically, and in Section 11.4 apply these results to obtain exponentially decaying tail bounds in the case where the errors decrease linearly.

We assume for simplicity that $\alpha_k \equiv 1/L$. We make the following blanket assumptions about f. First, the solution set $\mathcal{S}^*$ of (1.3.2) is nonempty. For all x and y, there exist positive constants L and $\tau \geq 1$ such that

$$\|\nabla f(y) - \nabla f(x)\| \leq L \, \|y - x\|, \tag{11.0.3a}$$

$$\min_{\bar{x} \in \mathcal{S}^*} \|x - \bar{x}\| \leq \tau \, \|x - \mathbf{prox}_{1/L}(x - \tfrac{1}{L} \nabla f(x))\| \qquad \forall x \in \operatorname{dom}(g) \tag{11.0.3b}$$

Assumption (11.0.3a) asserts the Lipschitz continuity of the gradient of f. Assumption (11.0.3b) is an error bound on the generalized residual. This generalized residual has been explored in local contexts in Tseng and Yun [TY09] and Luo and Tseng [LT93a]; for simplicity our assumption is stronger, however, requiring the bound to be global.

τ can be seen as a nonsmooth proxy for the condition number. If $g \equiv 0$ and f is strongly convex with parameter μ, then (11.0.3b) holds with $\tau = L/\mu$. More generally, if g is an indicator function on a polyhedral set and f is strongly convex, this bound holds globally, with parameter $\tau = (L+1)/\mu$ [Pan87, Theorem 3.1]. Recently, a global version of this error bound has been developed for non-strongly convex functions which degrades with the size of the neighborhood [WL14, Theorem 18]. We can use such a bound if the function g is an indicator over a polyhedral set, and f can be written in the form

$$f(x) = r(Ax) + b^T x$$

for any A, b, and r is strongly convex.

11.1. Proximal Gradient with Error

If conditions (11.0.3) hold then we can prove the following linear rate of convergence

Lemma 11.1.1. Let $\pi_k := [f+g](x_k) - \min_x [f+g](x)$. Then after k iterations of algorithm (1.3.6),

$$\pi_k \leq \rho^k \pi_0 + \frac{1}{\vartheta} \sum_{i=0}^{k-1} \rho^{k-1-i} \|e_i\|^2,$$

where

$$\rho = 1 - \frac{1}{1 + 40\tau^2} \in (0,1) \qquad \text{and} \qquad \vartheta = L \cdot \left(\frac{1}{40\tau^2} + 1 \right) > 0.$$

The proof of this result follows the template laid out by Luo and Tseng [LT93b, Theorem 3.1], modified to keep the error term e_k explicit. So [So13] also provides a similar derivation for the case where $g \equiv 0$, in which case it seems possible to obtain tighter constants ρ and ϑ. If additionally $\|e_k\| = 0$, then the result reduces to the well-known fact that steepest descent decreases the objective value linearly. The convergence rate, as expected, is a function of the condition number. We note that the constants are invariant to scalings of $f + g$.

Lemma 11.1.2 (Three-point property with error). For all $y \in \operatorname{dom}(g)$,

$$g(y) \geq g(x_{k+1}) + (\nabla f(x_k) + e_k)^T x_{k+1} - y + \frac{L}{2}\|x_{k+1} - x_k\|^2 + \frac{L}{2}\|y - x_{k+1}\|^2 - \frac{L}{2}\|y - x_k\|^2.$$

PROOF. Let $\psi_k(x) := g(x) + f(x_k) + (\nabla f(x_k) + e_k)^T(x - x_k) + \frac{L}{2}\|x - x_k\|^2$. Because ψ_k is strongly convex,

$$\psi_k(y) \geq \psi_k(x) + q^T y - x + \frac{L}{2}\|y - x\|^2 \quad \text{for all } x, y \text{ and all } q \in \partial\psi_k(x).$$

Choose $x = x_{k+1} := \operatorname{argmin} \phi_k(x)$. Because $0 \in \partial\psi_k(x_{k+1})$, we have

$$\psi_k(y) \geq \psi_k(x_{k+1}) + \frac{L}{2}\|y - x_{k+1}\|^2,$$

which, after simplifying, yields the required result. □

Lemma 11.1.3. Let $\mathcal{S}^*$ be the solution set, and let $\bar{x}_k$ be the projection of x_k onto $\mathcal{S}^*$. Then

$$\|x_k - \bar{x}_k\| \leq \tau\|x_k - x_{k+1}\| + \frac{\tau}{L}\|e_k\|; \tag{11.1.1a}$$

$$\|x_k - \bar{x}_k\|^2 \leq 2\tau^2\|x_k - x_{k+1}\|^2 + \frac{5}{4}(\tau^2/L^2)\|e_k\|^2; \tag{11.1.1b}$$

$$\|x_{k+1} - \bar{x}_k\| \leq (1+\tau)\|x_k - x_{k+1}\| + \frac{\tau}{L}\|e_k\|; \tag{11.1.1c}$$

$$\|x_{k+1} - \bar{x}_k\|^2 \leq \frac{1}{2}[2 + 5\tau + 3\tau^2]\|x_k - x_{k+1}\|^2 + \tfrac{1}{2L^2}[3\tau^2 + \tau]\|e_k\|^2. \tag{11.1.1d}$$

PROOF. For all k,

$$\begin{aligned}
\|x_k - \bar{x}_k\| &\overset{(i)}{\leq} \tau\|x_k - [x_k - \tfrac{1}{L}\nabla f(x_k)]_+\| \\
&\leq \tau\|x_k - x_{k+1}\| + \tau\|x_{k+1} - [x_k - \tfrac{1}{L}\nabla f(x_k)]_+\| \\
&= \tau\|x_k - x_{k+1}\| + \tau\|[x_k - \tfrac{1}{L}(\nabla f(x) + e_k)]_+ - [x_k - \tfrac{1}{L}\nabla f(x_k)]_+\| \\
&\overset{(ii)}{\leq} \tau\|x_k - x_{k+1}\| + \tfrac{\tau}{L}\|e_k\|,
\end{aligned}$$

where (i) follows from Assumption (11.0.3b) and (ii) follows from the nonexpansiveness of the proximal operator.

Part (11.1.1b)**.** Square both sides of (11.1.1a) and then apply the inequality

$$ab \leq \frac{a^2}{2\alpha} + \frac{\alpha b^2}{2}, \quad \forall\alpha > 0, \tag{11.1.2}$$

to bound the cross terms:

$$\begin{aligned}
\|x_k - \bar{x}_k\|^2 &\le \tau^2\|x_k - x_{k+1}\|^2 + (\tau/L)^2\|e_k\|^2 + (\tau^2/L)\|x_k - x_{k+1}\|\|e_k\| \\
&\le \left(\tau^2 + \tfrac{\tau^2\alpha}{2L}\right)\|x_k - x_{k+1}\|^2 + \left(\tfrac{\tau^2}{L^2} + \tfrac{\tau^2}{2L\alpha}\right)\|e_k\|^2 \qquad (\forall \alpha > 0) \\
&\le 2\tau^2\|x_k - x_{k+1}\|^2 + \tfrac{5}{4}(\tau^2/L^2)\|e_k\|^2.
\end{aligned}$$

Part (11.1.1c)**.** Use the triangle inequality and (11.1.1a): $\|x_{k+1}-\bar{x}_k\| \le \|x_{k+1}-x_k\|+\|x_k-\bar{x}_k\| \le (1+\tau)\|x_k - x_{k+1}\| + (\tau/L)\|e_k\|$.

Part (11.1.1d)**.** Square both sides above, and use the same technique used in Part (11.1.1b) to bound the cross-terms: $\|x_{k+1} - \bar{x}_k\|^2 \le \frac{1}{2}(2 + 5\tau + 3\tau^2)\|x_k - x_{k+1}\|^2 + \frac{1}{2L^2}(3\tau^2 + \tau)\|e_k\|^2$. □

Lemma 11.1.4 (Sufficient decrease)**.** For all k,

$$\pi_{k+1} \le \left(1 - \frac{1}{1 + 40\tau^2}\right)\pi_k + \frac{1}{L} \cdot \frac{40\tau^2}{1 + 40\tau^2}\|e_k\|^2.$$

Proof. First, specialize Lemma 11.1.2 with $y = x_k$:

$$g(x_{k+1}) \le g(x_k) - (\nabla f(x_k) + e_k)^T(x_{k+1} - x_k) - L\|x_{k+1} - x_k\|^2. \tag{11.1.3}$$

Then,

$$\begin{aligned}
h(x_{k+1}) &\overset{(i)}{\le} f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + g(x_{k+1}) \\
&\overset{(ii)}{\le} f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + g(x_k) \\
&\qquad - (\nabla f(x_k) + e_k)^T(x_{k+1} - x_k) - L\|x_{k+1} - x_k\|^2 \\
&= h(x_k) - e_k^T(x_{k+1} - x_k) - \frac{L}{2}\|x_k - x_{k+1}\|^2 \\
&\le h(x_k) + \frac{1}{2\alpha}\|e_k\|^2 + \left(\frac{\alpha}{2} - \frac{L}{2}\right)\|x_k - x_{k+1}\|^2,
\end{aligned}$$

where (i) uses Assumption (11.0.3a) and (ii) uses the (11.1.3). Choose $\alpha = L/2$ and rearrange terms to obtain the required result. □

We now proceed with the proof of Lemma 11.1.1. PROOF. Let $\bar{x}_k$ be the projection of x_k onto the solution set $\mathcal{S}^*$. By the mean value theorem,

$$f(x_{k+1}) - f(\bar{x}_k) = \nabla f(\xi)^T(x_{k+1} - \bar{x}_k). \tag{11.1.4}$$

From Lemma 11.1.2, we have

$$\begin{aligned} g(x_{k+1}) - g(\bar{x}_k) &\le -(\nabla f(x_k) + e_k)^T(x_{k+1} - \bar{x}_k) - \frac{L}{2}\|x_{k+1} - x_k\|^2 - \frac{L}{2}\|\bar{x}_k - x_{k+1}\|^2 + \frac{L}{2}\|\bar{x}_k - x_k\|^2 \\ &\le -(\nabla f(x_k) + e_k)^T(x_{k+1} - \bar{x}_k) + \frac{L}{2}\|\bar{x}_k - x_k\|^2. \end{aligned} \tag{11.1.5}$$

Also note that

$$\begin{aligned} (\nabla f(\xi) - \nabla f(x_k))^T(x_{k+1} - \bar{x}_k) &\le \|\nabla f(\xi) - \nabla f(x_k)\|\|x_{k+1} - \bar{x}_k\| \\ &\overset{(i)}{\le} L\|\xi - x_k\|\|x_{k+1} - \bar{x}_k\| \\ &\le L[\|x_{k+1} - x_k\| + \|x_k - \bar{x}_k\|] \cdot \|x_{k+1} - \bar{x}_k\| \\ &\le [L(1+\tau)\|x_k - x_{k+1}\| + \tau\|e_k\|] \cdot [(1+\tau)\|x_k - x_{k+1}\| + \tfrac{\tau}{L}\|e_k\|] \\ &= L(1+\tau)^2\|x_k - x_{k+1}\|^2 + 2[\tau(1+\tau)]\|x_k - x_{k+1}\|\|e_k\| + \tau^2/L\|e_k\|^2 \\ &\le [L(1+\tau)^2 + \tfrac{1}{\alpha}\tau(1+\tau)]\|x_k - x_{k+1}\|^2 + [\tau^2/L + \alpha\tau(1+\tau)]\|e_k\|^2 \\ &\le L(1 + 3\tau + 2\tau^2)\|x_k - x_{k+1}\|^2 + \tfrac{1}{L}(2\tau^2 + \tau)\|e_k\|^2, \end{aligned}$$

where (i) follows from (11.0.3a). In the steps which follow, we apply the relevant inequalities in Lemma 11.1.3, group terms, bound every cross term using (11.1.2), and repeat the process until we reach the final result:

$$\begin{aligned} h(x_{k+1}) - h(\bar{x}_k) &\overset{(i)}{\le} (\nabla f(\xi) - \nabla f(x_k))^T(x_{k+1} - \bar{x}_k) - e_k^T(x_{k+1} - \bar{x}_k) + \frac{L}{2}\|\bar{x}_k - x_k\|^2 \\ &\le L(1 + 3\tau + 2\tau^2)\|x_k - x_{k+1}\|^2 + \tfrac{1}{L}(2\tau^2 + \tau)\|e_k\|^2 \\ &\quad + \frac{\alpha}{2}\|e_k\|^2 + \tfrac{1}{2\alpha}\|x_{k+1} - \bar{x}_k\|^2 + \frac{L}{2}\|\bar{x}_k - x_k\|^2 \qquad \forall \alpha > 0 \\ &\le L(1 + 3\tau + 2\tau^2)\|x_k - x_{k+1}\|^2 + \tfrac{1}{L}(2\tau^2 + \tau)\|e_k\|^2 \\ &\quad + \frac{\alpha}{2}\|e_k\|^2 + \tfrac{1}{4\alpha}[2 + 5\tau + 3\tau^2]\|x_k - x_{k+1}\|^2 \end{aligned}$$

$$
\begin{aligned}
&+ \tfrac{1}{4L^2\alpha}[3\tau^2+\tau]\|e_k\|^2 + L\tau^2\|x_k - x_{k+1}\|^2 + \tfrac{5L}{8}(\tau^2/L^2)\|e_k\|^2 \\
&\leq \left(L(1+3\tau+2\tau^2) + \tfrac{1}{4\alpha}[2+5\tau+3\tau^2] + L\tau^2\right)\|x_k - x_{k+1}\|^2 + \\
&\quad \left(\tfrac{1}{L}(2\tau^2+\tau) + \frac{\alpha}{2} + \tfrac{1}{4L^2\alpha}[3\tau^2+\tau] + \tfrac{5L}{8}(\tau^2/L^2)\right)\|e_k\|^2 \\
&\overset{(ii)}{\leq} 10L\tau^2\|x_k - x_{k+1}\|^2 + \tfrac{1}{L}10\tau^2\|e_k\|^2 \\
&\overset{(iii)}{\leq} 40\tau^2[h(x_k) - h(x_{k+1})] + (4/L^2 + \tfrac{1}{L}10\tau^2)\|e_k\|^2 \\
&\leq 40\tau^2[h(x_k) - h(x_{k+1})] + \tfrac{1}{L}40\tau^2\|e_k\|^2.
\end{aligned}
$$

In the steps above, (i) follows by adding inequalities (11.1.4) and (11.1.5). Also, we make use of Lemma 11.1.3 to bound all stray terms in terms of $\|x_k - x_{k+1}\|^2$ and $\|e_k\|^2$, and Equation (11.1.2) to bound the cross-terms. In (ii) we make use of the assumption that $\tau \geq 1$ and set $\alpha = 1/L$. Finally, in (iii) we make use of Lemma 11.1.4 to transition from a bound on the distance between successive iterates x_k to differences in successive values $h(x_k)$. Rearranging terms, we get

$$(1+40\tau^2)h(x_{k+1}) - (1+40\tau^2)h(\bar{x}_k) \leq 40\tau^2(h(x_k) - h(\bar{x}_k)) + \frac{1}{L}40\tau^2\|e_k\|^2,$$

which is true if and only if the desired result holds:

$$\pi_{k+1} \leq \left(1 - \frac{1}{1+40\tau^2}\right)\pi_k + \frac{1}{L}\cdot\frac{40\tau^2}{1+40\tau^2}\|e_k\|^2.$$

□

This bound is of immediate utility when we have deterministic bounds on the errors $\|e_k\|^2$, e.g. (10.3.2). And we can exploit the linearity of expectation to obtain a bound on expectation,

$$\mathbf{E}\,\pi_k \leq \rho^k\pi_0 + \frac{1}{\vartheta}\sum_{i=0}^{k-1}\rho^{k-1-i}\,\mathbf{E}\,\|e_i\|^2,$$

which by Markov's Inequality implies convergence in probability.

Example 11.1.5 (Gradient descent with independent Gaussian noise, part I)**.** Let $e_k \sim N(0, \sigma^2 I)$. Because $\|e_k\|^2$ is a sum of n independent Gaussians, it follows a chi-squared distribution with mean

$\mathbf{E}\|e_k\|^2 = n\sigma^2$. Therefore,

$$\mathbf{E}\pi_k - \rho^k\pi_0 \le \frac{1}{\vartheta}\sum_{i=0}^{k-1}\rho^{k-1-i}\mathbf{E}\|e_i\|^2 = \frac{n\sigma^2}{\vartheta}\sum_{i=0}^{k-1}\rho^{k-1-i}. \tag{11.1.6}$$

Take the limit inferior of both sides of (11.1.6), and note that $\lim_{k\to\infty}\sum_{i=0}^{k-1}\rho^{k-1-i} = 1/(1-\rho)$. Use the values of the constants in Lemma 11.1.1 to obtain the bound

$$\mathbf{E}\liminf_{k\to\infty}\pi_k \le \liminf_{k\to\infty}\mathbf{E}\pi_k \le \frac{20\tau^2}{L}\,n\sigma^2,$$

where the first inequality follows from the application of Fatou's Lemma [RF10, Ch. 4]. Hence, even though $\lim_{k\to\infty}\pi_k$ may not exist, we can still provide a lower bound on the distance to optimality that is proportional to the variance of the error term.

11.2. Probabilistic bounds for gradient descent with random error

The first bound (Section 11.2.1) that we develop makes no assumption on the relation of the gradient errors between iterations, i.e., the error sequence may or may not be history dependent, and we thus refer to this as a generic error sequence. The second bound (Section 11.2.2) makes the stronger assumption about the relationship of the errors between iterations. The second bound (Section 11.2.2) makes the stronger assumption about the relationship of the errors between iterations.

Let $R_k := \sum_{i=1}^{k=1}\rho^i$, where $\rho < 1$ is a constant specified in Lemma 11.1.1, and $\sigma(e_1, e_2, \ldots, e_k)$ be the σ-algebra generated by the sequence of errors e_i.

11.2.1. Generic error sequence. Our first exponential tail bounds are defined in terms of the moment-generating function

$$\gamma_k(\theta) := \mathbf{E}\exp(\theta\|e_k\|^2)$$

of the error norms $\|e_k\|^2$. We make the convention that $\gamma_k(\theta) = +\infty$ for $\theta \notin \mathrm{dom}\gamma_k$.

Theorem 11.2.1 (Tail bound for generic errors)**.** For algorithm (1.3.6),

$$\Pr(\pi_k - \rho^k\pi_0 \ge \epsilon) \le \inf_{\theta>0}\left\{\frac{\exp(-\theta\vartheta\epsilon/R_k)}{R_k}\sum_{i=0}^{k-1}\rho^{k-1-i}\gamma_i(\theta)\right\}. \tag{11.2.1a}$$

If $\gamma_k \equiv \gamma$ for all k (i.e., the error norms $\|e_k\|^2$ are identically distributed), then the bound simplifies to

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \inf_{\theta > 0} \left\{ \exp(-\theta\vartheta\epsilon / R_k)\gamma(\theta) \right\}. \tag{11.2.1b}$$

Proof. By the definition of R_k, $\left(\sum_{i=0}^{k-1} \rho^{k-1-i}\right)/R_k = 1$. Thus, for $\theta > 0$,

$$\begin{aligned}\mathbf{E}\exp\left(\theta \sum_{i=0}^{k-1} \rho^{k-1-i}\|e_i\|^2\right) &= \mathbf{E}\exp\left(\sum_{i=0}^{k-1} \frac{\rho^{k-1-i}}{R_k}\theta R_k \|e_i\|^2\right) \\ &\overset{(i)}{\leq} \mathbf{E}\sum_{i=0}^{k-1} \frac{\rho^{k-1-i}}{R_k}\exp(\theta R_k\|e_i\|^2) \overset{(ii)}{=} \frac{1}{R_k}\sum_{i=0}^{k-1}\rho^{k-1-i}\gamma_i(\theta R_k),\end{aligned}$$

where (i) follows from the convexity of $\exp(\cdot)$, and (ii) follows from the linearity of the expectation operator and the definition of γ_i. Together with Markov's inequality, the above implies that for all $\theta > 0$,

$$\begin{aligned}\Pr\left(\sum_{i=0}^{k-1}\rho^{k-1-i}\|e_i\|^2 \geq \epsilon\right) &= \Pr\left(\exp\left[\theta\sum_{i=0}^{k-1}\rho^{k-1-i}\|e_i\|^2\right] \geq \exp(\theta\epsilon)\right) \\ &\leq \exp(-\theta\epsilon)\mathbf{E}\exp\left(\theta\sum_{i=0}^{k-1}\rho^{k-1-i}\|e_i\|^2\right) \\ &\leq \frac{\exp(-\theta\epsilon)}{R_k}\sum_{i=0}^{k-1}\rho^{k-1-i}\gamma_i(\theta R_k). \qquad (11.2.2)\end{aligned}$$

This inequality, together with Lemma 11.1.1, implies that for all $\theta > 0$,

$$\Pr\left(\pi_k - \rho^k\pi_0 \geq \epsilon\right) \leq \Pr\left(\frac{1}{\vartheta}\sum_{i=0}^{k-1}\rho^{k-1-i}\|e_i\|^2 \geq \vartheta\epsilon\right) \leq \frac{\exp(-\theta\vartheta\epsilon)}{R_k}\sum_{i=0}^{k-1}\rho^{k-1-i}\gamma_i(\theta R_k),$$

where we use the elementary fact that $\Pr(X \geq \epsilon) \leq \Pr(Y \geq \epsilon)$ if $X \leq Y$ almost surely. Redefine θ as θR_k, and take the infimum of the right-hand side over $\theta > 0$, which gives the required inequality (11.2.1a). The simplified bound (11.2.1b) follows directly from the definition of R_k. $\square$

When the errors are identically distributed, there is an intriguing connection between the tail bounds described in Theorem 11.2.1 and the convex conjugate of the cumulant-generating function of that distribution, i.e., $(\log\circ\gamma)^*$.

Theorem 11.2.2 (Tail bound for identically-distributed errors)**.** Suppose that the error norms $\|e_k\|^2$ are identically distributed. Then for algorithm (1.3.6),

$$\log \Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq - [\log \gamma(\cdot)]^* (\vartheta\epsilon/R_k).$$

Proof. Take the log of both sides of (11.2.1b) to get

$$\log \Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \log \inf_{\theta>0} \{\exp(-\theta\vartheta\epsilon/R_k)\, \gamma(\theta)\} = -\sup_{\theta>0} \{(\vartheta\epsilon/R_k)\theta - \log \gamma(\theta)\},$$

which we recognize as the negative of the conjugate of $\log \circ \gamma$ evaluated at $\vartheta\epsilon/R_k$. □

Note that these bounds are invariant with regard to scaling, in the sense that if the objective function f is scaled by some $\alpha > 0$, then the bounds hold for $\alpha\epsilon$.

The following example illustrates an application of this tail bound to the case in which the errors follow a simple distribution with a known moment-generating function.

Example 11.2.3 (Gradient descent with independent Gaussian noise, part II)**.** As in Example 11.1.5, let $e_k \sim N(0, \sigma^2 I)$. Then e_k is a scaled chi-squared distribution with moment-generating function

$$\gamma_k(\theta) = (1 - 2\sigma^2\theta)^{-n/2}, \quad \theta \in \left[0, \frac{1}{2\sigma^2}\right).$$

Note that

$$[\log \gamma(\cdot)]^*(\mu) = \frac{\mu - n\sigma^2}{2\sigma^2} + \frac{n}{2} \log(n\sigma^2/\mu) \text{for} \mu > n\sigma^2.$$

We can then apply Corollary 11.2.2 to this case to deduce the bound

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \left(\frac{\exp(1)}{n} \cdot \frac{\vartheta\epsilon}{\sigma^2 R_k}\right)^{n/2} \exp\left(-\frac{\vartheta\epsilon}{2\sigma^2 R_k}\right) \quad \text{for} \quad \epsilon > \frac{n\sigma^2 R_k}{\vartheta}.$$

The bound can be further simplified by introducing an additional perturbation $\delta > 0$ that increases the base of the exponent:

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) = \mathcal{O}\left[\exp\left(-\delta \frac{\vartheta\epsilon}{2\sigma^2 R_k}\right)\right] \quad \text{for all} \quad \delta \in [0, 1), \tag{11.2.3}$$

which highlights the exponential decrease of the bound in terms of ϵ.

11.2.2. Unconditionally bounded error sequence. In contrast to the previous section, we now assume that there exists a deterministic bound on the conditional expectation $\mathbf{E}\left[\exp(\theta\|e_k\|^2) \mid \mathcal{F}_{k-1}\right]$. We say that this bound holds unconditionally because it holds irrespective of the history of the error sequence.

Lemma 11.2.4. Assume that $\mathbf{E}\left[\exp(\theta\|e_k\|^2) \mid \mathcal{F}_{k-1}\right]$ is finite over $[0, \sigma)$, for some $\sigma > 0$. Therefore there exists, for each k, a deterministic function $\bar{\gamma}_k : \mathbf{R}_+ \to \mathbf{R}_+ \cup \{\infty\}$ such that

$$\bar{\gamma}_k(0) = 1 \qquad \text{and} \qquad \mathbf{E}\left[\exp(\theta\|e_k\|^2) \mid \mathcal{F}_{k-1}\right] \le \bar{\gamma}_k(\theta).$$

(Thus, the bound is tight at $\theta = 0$.)

The existence of such a function in fact implies a bound on the moment-generating function of $\|e_k\|^2$. In particular,

$$\gamma_k(\theta) := \mathbf{E}\exp(\theta\|e_k\|^2) = \mathbf{E}\left[\mathbf{E}\left[\exp(\theta\|e_k\|^2) \mid \mathcal{F}_{k-1}\right]\right] \le \mathbf{E}\bar{\gamma}_k(\theta) = \bar{\gamma}_k(\theta). \tag{11.2.4}$$

The converse, however, is not necessarily true. To see this, consider the case in which the errors $e_1, \ldots, e_{k-1}$ are independent Bernoulli-distributed random variables, and e_k is a deterministic function of all the previous errors, e.g., $\Pr(e_i = 0) = \Pr(e_i = 1) = 1/2$ for $i = 1, \ldots, k-1$, and the error on the last iteration is completely determined by the previous errors:

$$e_k = \begin{cases} 1 & \text{if } e_1 = e_2 = \cdots = e_{k-1}, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $\Pr(e_k = 1) = (1/2)^{k-1}$ and $\Pr(e_k = 0) = 1 - (1/2)^{k-1}$, and the moment-generating function of e_k is $\gamma_k(\theta) = 1 - 2^{1-k}(1 + \exp\theta)$. Then,

$$\mathbf{E}[\exp(\theta e_k^2) \mid e_1, \ldots, e_{k-1}] = \begin{cases} \exp\theta & \text{if } e_1 = e_2 = \cdots = e_{k-1}, \\ 1 & \text{otherwise,} \end{cases}$$

whose tightest deterministic upper bound is $\bar{\gamma}_k(\theta) = \exp\theta$. However, $\bar{\gamma}_k(\theta) \ge \gamma_k(\theta)$ for all $\theta \ge 0$. The following result is analogous to Theorem 11.2.1.

Theorem 11.2.5 (Tail bounds for unconditionally bounded errors)**.** Suppose that Assumption 11.2.4 holds. Then for algorithm (1.3.6),

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \inf_{\theta > 0} \left\{ \exp(-\theta\vartheta\epsilon) \prod_{i=0}^{k-1} \bar{\gamma}_i(\theta\rho^{k-i-1}) \right\}.$$

Proof. The proof follows the same outline as many martingale-type inequalities [Azu67, CL06]. We obtain the following relationships:

$$\begin{aligned}
\mathbf{E} \exp\left[\theta \sum_{i=0}^{k-1} \rho^{k-1-i} \|e_i\|^2\right] &\overset{(i)}{=} \mathbf{E}\left[\mathbf{E}\left[\exp\left[\theta \sum_{i=0}^{k-1} \rho^{k-1-i}\|e_i\|^2\right] \middle| \mathcal{F}_{k-2}\right]\right] \\
&= \mathbf{E}\left[\mathbf{E}\left[\exp\left[\theta\rho^0\|e_{k-1}\|^2 + \theta\sum_{i=0}^{k-2}\rho^{k-1-i}\|e_i\|^2\right] \middle| \mathcal{F}_{k-2}\right]\right] \\
&\overset{(ii)}{=} \mathbf{E}\left[\exp\left[\theta\sum_{i=0}^{k-2}\rho^{k-1-i}\|e_i\|^2\right] \mathbf{E}\left[\exp\left(\theta\|e_{k-1}\|^2\right) \middle| \mathcal{F}_{k-2}\right]\right] \\
&\overset{(iii)}{\leq} \mathbf{E}\left[\exp\left[\theta\sum_{i=0}^{k-2}\rho^{k-i-1}\|e_i\|^2\right]\right] \bar{\gamma}_{k-1}(\theta) \\
&\overset{(iv)}{\leq} \prod_{i=0}^{k-1} \bar{\gamma}_i(\theta\rho^{k-i-1}),
\end{aligned}$$

where (i) follows from the law of total expectations, i.e., $\mathbf{E}_Y[\mathbf{E}[X|Y]] = \mathbf{E}[X]$; (ii) follows from the observation that the random variable $\exp(\theta\sum_{i=0}^{k-2}\rho^{k-1-i}\|e_i\|^2)$ is a deterministic function of $e_0, \ldots, e_{k-2}$, and hence is measurable with respect to $\mathcal{F}_{k-1}$ and can be factored out of the expectation; (iii) uses Assumption 11.2.4; and to obtain (iv) we simply repeat the process recursively.

Thus, we now have a bound on the moment-generating function of the discounted sum of errors $\theta\sum_{i=0}^{k-1}\rho^{k-1-i}\|e_i\|^2$, and we can continue by using the same approach used to derive (11.2.2). The remainder of the proof follows that of Theorem 11.2.1, except that the sums over $i = 0, \ldots, k$ are replaced by products over that same range. □

In an application where both γ_k and $\bar{\gamma}_k$ are available, it is not true in general that either of the bounds obtained in Theorems 11.2.1 and 11.2.5 are tighter than the other. When only a bound $\bar{\gamma}_k$ that satisfies Assumption 11.2.4 is available, however, (which is the case in the sampling application we shall describe) we could leverage (11.2.4) and apply Theorem 11.2.1 to obtain a valid bound in terms of $\bar{\gamma}_k$ by simply substituting it for γ_k. However, as shown below, in this case it is better to apply Theorem 11.2.5 because it yields a uniformly better bound:

$$\Pr\left(\pi_k - \rho^k \pi_k \geq \epsilon\right) \leq \inf_{\theta>0}\left\{\exp\left(-\theta\vartheta\epsilon + \sum_{i=0}^{k-1}\log\bar{\gamma}_i\big(\theta\rho^{k-1-i}\big)\right)\right\}, \tag{11.2.5}$$

while Theorem 11.2.1 (with γ_k replaced by $\bar{\gamma}_k$) gives us

$$\Pr\left(\pi_k - \rho^k \pi_0 \geq \epsilon\right) \leq \inf_{\theta>0}\left\{\exp\left(-\theta\vartheta\epsilon + \log\left[\frac{1}{R_k}\sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\gamma}_i(\theta R_k)\right]\right)\right\}, \tag{11.2.6}$$

where we rescale θ by R_k. A direct comparison of the two bounds show that they only differ by one term:

$$\log\left[\frac{1}{R_k}\sum_{i=0}^{k-1}\rho^{k-i-1}\bar{\gamma}_i(\theta R_k)\right] \quad \text{vs.} \quad \sum_{i=0}^{k-1}\log\bar{\gamma}_i(\theta\rho^{k-1-i}).$$

Because $R_k = \sum_{i=0}^{k}\rho^{k-i-1}$, the term in the log on the left is a convex combination of the functions $\bar{\gamma}_i$. Therefore,

$$\begin{aligned}
\log\left[\frac{1}{R_k}\sum_{i=0}^{k-1}\rho^{k-i-1}\bar{\gamma}_i(\theta R_k)\right] &\overset{(i)}{\geq} \sum_{i=0}^{k-1}\frac{\rho^{k-1-i}}{R_k}\log\bar{\gamma}_i(\theta R_k)\\
&\overset{(ii)}{\geq} \sum_{i=0}^{k-1}\log\bar{\gamma}_i(\theta R_k\,\rho^{k-1-i}/R_k)\\
&= \sum_{i=0}^{k-1}\log\bar{\gamma}_i(\theta\rho^{k-1-i}),
\end{aligned}$$

where (i) is an application of Jensen's inequality and the concavity of log, and (ii) follows from the convexity of the cumulant generating function. It is then evident that (11.2.5) implies (11.2.6).

As with Corollary 11.2.2, by taking logs of both sides above, a connection can be made between our bound and the infimal convolution when $\bar{\gamma}$ is log-concave:

$$\log\Pr(\pi_k - \rho^k\pi_0 \geq \epsilon) \leq \left[\bigotimes_{i=0}^{k-1}[\log\bar{\gamma}_i(\,\cdot\,\rho^{k-i-1})]^*\right](\vartheta\epsilon/R_k),$$

where $\otimes$ denotes the infimal convolution operator.

Example 11.2.6 (Gradient descent with independent Gaussian noise, part III)**.** As in Example 11.2.3, let $e_k \sim N(0,\sigma^2 I)$. Because the errors e_k are independent, $\mathbf{E}\big[\exp(\theta\|e_k\|^2)\,|\,\mathcal{F}_{k-1}\big] =$

$\mathbf{E}\exp(\theta\|e_k\|^2) = \gamma_k(\theta)$, which satisfies Assumption 11.2.4 with $\bar{\gamma}_k(\theta) := \gamma_k(\theta)$. Apply Theorem 11.2.5 to obtain the bound

$$\Pr(\pi_k - \rho^k\pi_0 \geq \epsilon) \leq \inf_{\theta>0}\left\{\exp(-\theta\vartheta\epsilon)\cdot\prod_{i=0}^{k-1}(1-2\sigma^2\theta\rho^{k-1-i})^{-n/2}\right\}. \tag{11.2.7}$$

Apply Lemma B.1.2 to obtain

$$\Pr(\pi_k - \rho^k\pi_0 \geq \epsilon) \leq \left(\frac{\exp(1)}{n\alpha}\cdot\frac{\vartheta\epsilon}{\sigma^2}\right)^{\frac{n\alpha}{2}}\exp\left(-\frac{\vartheta\epsilon}{\sigma^2}\right) \quad \text{for} \quad \epsilon > \frac{n\alpha\sigma^2}{\vartheta},$$

where $\alpha = 1-(\log\rho)^{-1}$. We simplify the bound to obtain

$$\Pr(\pi_k - \rho^k\pi_0 \geq \epsilon) = \mathcal{O}\left[\exp\left(-\delta\cdot\frac{\vartheta\epsilon}{\sigma^2}\right)\right] \quad \text{for all} \quad \delta\in(0,1); \tag{11.2.8}$$

cf. (11.2.3).

As an aside, we note that we can easily accommodate correlated noise, i.e., $e_k \sim N(0,\Sigma^2)$ where Σ is an $n\times n$ positive definite matrix. The error $\|e_k\|^2$ then has the distribution of a sum of chi-squared random variables that are weighted according to the eigenvalues σ_j of Σ [Imh61]:

$$\|e_k\|^2 \sim \sum_{j=1}^{n}\sigma_j^2\chi_1^2,$$

and so the above tail bounds hold with $\sigma = \sigma_{\max}$.

The bounds obtained in Examples 11.2.3 and 11.2.6 illustrate the relative strengths of Theorems 11.2.1 and 11.2.5. Comparing (11.2.3) and (11.2.8), we see that the asymptotic bounds only differ by a factor of $1/R_k$. Hence, for large ϵ, the bound in Example 11.2.3 is uniformly weaker than the bound in Example 11.2.6. Note that this holds despite the simplification (i.e., Lemma B.1.2) used to simply (11.2.7).

11.3. From tail bounds to moment-generating bounds

Let $\mathcal{G}$ be a σ-algebra. Consider the exponential bound on the conditional probability [Kle08, Definition 8.11] of a sequence of univariate random variables X_i:

$$\Pr(X_i \geq \epsilon \mid \mathcal{G}) := \mathbf{E}[\mathbb{I}[X_i \geq \epsilon] \mid \mathcal{G}\,] \leq \exp(-\epsilon^2/\nu) \quad \text{for some} \quad \nu > 0. \tag{11.3.1}$$

In this section we show that this bound translates into a deterministic bound on the conditional moment-generating function

$$\mathbf{E}[\exp(\theta\|X\|^2) \mid \mathcal{G}],$$

where $X = (X_1, X_2, \ldots, X_n)$ is an n-vector. The subsidiary lemmas follow standard arguments [BLM13, Chapter 2], except for the requirement to condition on $\mathcal{G}$; hence, we rederive the required results.

Lemma 11.3.1 (Bounds on moments)**.** If (11.3.1) holds for some $\nu > 0$, then

$$\mathbf{E}[X_i^{2v} \mid \mathcal{G}] \leq v!\,\nu^v \qquad \text{for all} \qquad v = 0, 1, 2, \ldots.$$

Proof. We follow a similar argument to [BLM13, Theorem 2.1]. Use the substitution $\epsilon^{2v} = \tau$ to obtain

$$\Pr\left(Y^{2v} \geq \tau \mid \mathcal{G}\right) \leq \exp\left(-\tau^{1/v}/\nu\right).$$

Integrate to get

$$\mathbf{E}[Y^{2v} \mid \mathcal{G}] = \int_0^\infty \mathbf{E}[\mathbb{I}[Y^{2v} \geq \tau] \mid \mathcal{G}]\, d\tau \;\leq\; \int_0^\infty \exp(-\tau^{1/v}/\nu)\, d\tau \;=\; \Gamma(1+v)\nu^v \;=\; v!\nu^v,$$

where the first equality comes from the conditional layer-cake representation of positive random variables [Swa09]. □

With this result, we can translate the bound (11.3.1) into a bound on the moment-generating function of Y^2.

Lemma 11.3.2 (Bound on conditional MGF)**.** If (11.3.1) holds for some $\nu > 0$, then

$$\mathbf{E}[\exp(\theta Y^2) \mid \mathcal{G}] \le \frac{1}{1-\theta\nu} \quad \text{for} \quad \theta \in [0, 1/\nu).$$

Proof. Using the Taylor expansion of $\mathbf{E}[\exp\left(\theta Y^2\right) \mid \mathcal{G}]$,

$$\mathbf{E}[\exp\left(\theta Y^2\right) \mid \mathcal{G}] = \mathbf{E}\left[\sum_{i=0}^{\infty} \theta^i \frac{Y^{2i}}{i!} \,\middle|\, \mathcal{G}\right] \overset{(i)}{=} \sum_{i=0}^{\infty} \theta^i \frac{\mathbf{E}[Y^{2i} \mid \mathcal{G}]}{i!} \overset{(ii)}{\le} \sum_{i=0}^{\infty} \theta^i \frac{i!\nu^i}{i!} = \sum_{i=0}^{\infty} (\theta\nu)^i = \frac{1}{1-\theta\nu}.$$

Equality (i) is obtained via the conditional monotone convergence theorem [Wil91, Theorem 9.7e], which allows us to exchange limits and conditional expectations; inequality (ii) is obtained using Lemma 11.3.1. □

We now generalize this last result to the case in which X is a random n-vector.

Theorem 11.3.3 (From tail bounds to moment-generating bounds)**.** Suppose X is a random n-vector for which the tail bound (11.3.1) holds for each i for some $\nu > 0$. Then

$$\mathbf{E}[\exp(\theta\|X\|^2) \mid \mathcal{G}] \le \frac{1}{1-\theta\nu n} \quad \text{for} \quad \theta \in [0, 1/\nu n).$$

Proof. From Lemma 11.3.2,

$$\mathbf{E}\left[\exp\left(\theta n [X]_i^2\right) \mid \mathcal{G}\right] \le \frac{1}{1-\theta n\nu}. \tag{11.3.2}$$

The following inequalities hold:

$$\begin{aligned}
\mathbf{E}\left[\exp\left(\theta\|X\|^2\right) \middle| \mathcal{G}\right] &= \mathbf{E}\left[\exp\left(\theta \sum_{i=1}^{n} [X]_i^2\right) \middle| \mathcal{G}\right] \\
&= \mathbf{E}\left[\exp\left(\theta n \sum_{i=1}^{n} \frac{1}{n} [X]_i^2\right) \middle| \mathcal{G}\right] \\
&\overset{(i)}{\le} \mathbf{E}\left[\sum_{i=1}^{n} \frac{1}{n} \exp\left(\theta n [X]_i^2\right) \middle| \mathcal{G}\right] = \sum_{i=1}^{n} \frac{1}{n} \mathbf{E}\left[\exp\left(\theta n [X]_i^2\right) \middle| \mathcal{G}\right] \overset{(ii)}{\le} \frac{1}{1-\theta n\nu},
\end{aligned}$$

where (i) follows from Jensen's inequality and (ii) follows from (11.3.2). □

11.4. Convergence rates for linearly decreasing errors

Section 11.2 describes tail bounds for (11.0.2) in terms of any available bound on the moment-generating function of the error e_k. A goal of this section is to show that an exponential tail bound on the error translates to an exponential tail bound on (11.0.2). Thus we consider the case where the tails on each component of e_k are exponentially decreasing (cf. Hypothesis 10.1.1.B below). We also consider two additional conditions on the error sequence, which illustrate the exponential tail bound's relative strength in the following hierarchy of assumptions.

11.4.1. Expectation-based and deterministic bounds.

Although our main goal is to derive tail bounds, it is useful to compare these against the expectation-based and deterministic bounds derived in [FS12, Theorem 3.3]. We give here a reformulation of these results, which rely on parts A and C of Hypothesis 10.1.1.

Theorem 11.4.1 (Bound in expectation)**.** If we use the schedule

$$s_k = A\text{-}\mathbf{sto\text{-}oracle}(x_k, \lambda\beta^k)$$

in the proximal iteration (11.0.1) for some $\lambda > 0$ and $\beta \in (0,1)$, then

$$\mathbf{E}\pi_k - \rho^k\pi_0 = \mathcal{O}(k[\max\{\beta,\rho\}]^k),$$

and if $\rho \neq \beta$, then the bound holds with $\mathcal{O}([\max\{\beta,\rho\}]^k)$. If we use $s_k = B\text{-}\mathbf{sto\text{-}oracle}(x_k, \lambda\beta^k)$ instead, then this result holds verbatim, except without the expectation operator.

Proof. For $\beta \leq \rho$, it follows from Lemma 11.1.1 and Hypothesis 10.1.1.A that

$$\mathbf{E}\pi_k - \rho^k\pi_0 \leq \frac{1}{\vartheta}\sum_{i=0}^{k-1}\rho^{k-i-1}\mathbf{E}\|e_i\|^2 \leq \frac{\lambda\rho^{k-1}}{\vartheta}\sum_{i=0}^{k-1}(\beta/\rho)^i \leq \frac{\lambda}{\vartheta}\rho^{k-1}k. \tag{11.4.1}$$

Similarly, for $\beta > \rho$,

$$\mathbf{E}\pi_k - \rho^k\pi_0 \leq \frac{\lambda\beta^{k-1}}{\vartheta}\sum_{i=0}^{k-1}(\rho/\beta)^i \leq \frac{\lambda}{\vartheta}\beta^{k-1}k. \tag{11.4.2}$$

We summarize these last two bounds in the single expression

$$\mathbf{E}\pi_k - \rho^k \pi_0 \le \frac{\lambda}{\vartheta} \max\{\beta,\rho\}^{k-1} k = \mathcal{O}(k[\max\{\beta,\rho\}]^k).$$

If $\beta \neq \rho$, then it follows from the second inequality in (11.4.1) and the first inequality in (11.4.2), and the summation formula for geometric series, that

$$\mathbf{E}\pi_k - \rho^k \pi_0 \le \frac{\lambda}{\vartheta} \max\{\beta,\rho\}^{k-1} \frac{1}{|\beta-\rho|} = \mathcal{O}(\max\{\beta,\rho\}^k). \tag{11.4.3}$$

If we are using a deterministic oracle instead, than the proof above proceeds verbatim, except that the expectation operator above can be removed. □

11.4.2. Tail bounds. The next result gives exponential tail bounds in terms of the iteration k, and the deviation ϵ from the linear rate of deterministic steepest descent.

Theorem 11.4.2 (Convergence rates)**.** If we use the schedule

$$s_k = C\textbf{-sto-oracle}(x_k, \lambda\beta^k) \tag{11.4.4}$$

in the proximal iteration (11.0.1) for some $\lambda > 0$ and $\beta \in (0,1)$, then for all k for some $\beta \in (0,1)$ and $\lambda > 0$,

$$\Pr(\pi_k - \rho^k\pi_0 \ge \epsilon) = \mathcal{O}\left(\exp\left[-\frac{\epsilon}{\max\{\beta,\rho\}^k}\cdot\zeta\right]\right)$$

where ζ depends on λ, τ and n.

Proof. The bound on the high probability stochastic oracle is uniform for all x, and hence applies regardless of error history sequence. Thus by Theorem 11.3.3 the conditioned moment-generating function of $\|e_k\|^2$ is bounded:

$$\mathbf{E}[\exp(\theta\|e_k\|^2) \mid \mathcal{F}_{k-1}] \le \frac{1}{1-\theta n U_k} \quad \text{for} \quad \theta \in \left[0, \frac{1}{nU_k}\right). \tag{11.4.5}$$

Define

$$\alpha_1 = \max_k \rho^{k-i-1} n U_k \qquad \text{and} \qquad \alpha_2 = \max\{\beta,\rho\}.$$

We can now use Theorem 11.2.5, where we identify $\bar{\gamma}$ with the bound in (11.4.5) (and define $\bar{\gamma}(\theta) = \infty$ outside of the required interval), to obtain the tail bound

$$\begin{aligned}\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) &\overset{(i)}{\leq} \inf_{\theta \in [0, 1/\alpha_1)} \left\{ \frac{\exp(-\theta \vartheta \epsilon)}{\prod_{i=0}^{k-1} \left(1 - \theta n U_k \rho^{k-i-1}\right)} \right\} \\ &\overset{(ii)}{\leq} \inf_{\theta \in [0, 1/\alpha_1)} \left\{ \frac{\exp(-\theta \vartheta \epsilon)}{\prod_{i=0}^{k-1} (1 - \theta n \lambda \beta^i \rho^{k-i-1})} \right\} \\ &\overset{(iii)}{=} \inf_{\theta \in [0, 1/\alpha_1)} \left\{ \frac{\exp(-\theta \vartheta \epsilon)}{\prod_{i=0}^{k-1} (1 - \theta n \lambda \alpha_2^{k-1} \min\{\beta/\rho,\, \rho/\beta\}^i)} \right\}, \end{aligned} \tag{11.4.6}$$

where (i) follows from the definition of α_1, (ii) follows from Definition 10.0.1, part (B) and our schedule 11.4.4, and (iii) follows from the definition of α_2.

Define $\alpha_3 = 1 + 1/\log(1/\min\{\beta/\rho,\, \rho/\beta\}) = 1 + 1/\left|\log \beta - \log \rho\right|$, and apply Lemma B.1.4 to (11.4.6) to obtain, for all $\epsilon \geq \alpha_3 \alpha_2^{k-1} n\lambda/\vartheta$,

$$\Pr\left(\pi_k - \rho^k \pi_0 \geq \epsilon\right) \leq \left(\frac{\exp(1)}{\alpha_3} \cdot \frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right)^{\alpha_3} \exp\left(-\frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right). \tag{11.4.7}$$

Next, note that $\min\{\beta/\rho,\, \rho/\beta\} \leq 1$, and so from (11.4.6), for all $\epsilon \geq k \alpha_2^{k-1} n\lambda/\vartheta$,

$$\begin{aligned}\Pr\left(\pi_k - \rho^k \pi_0 \geq \epsilon\right) &\leq \inf_{\theta \in [0, 1/\alpha_1)} \left\{ \frac{\exp(-\theta \vartheta \epsilon)}{(1 - \theta n \lambda \alpha_2^{k-1} \vartheta)^k} \right\} \\ &\overset{(i)}{\leq} \left(\frac{\exp(1)}{k} \cdot \frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right)^{k} \exp\left(-\frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right), \end{aligned} \tag{11.4.8}$$

where (i) follows from Lemma B.1.4. Let $\bar{\alpha}_k := \min\{\alpha_3, k\}$. Inequalities (11.4.7) and (11.4.8) can be expressed together, for all $\epsilon \geq \bar{\alpha}_k \alpha_2^{k-1} n\lambda/\vartheta$, as

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \left(\frac{\exp(1)}{\bar{\alpha}_k} \cdot \frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right)^{\bar{\alpha}_k} \exp\left(-\frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} \right). \tag{11.4.9}$$

Consider the case in which the $\mathcal{O}$ depends asymptotically only on ϵ. Then

$$\Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \mathcal{O}\left[\exp\left(-\delta \cdot \frac{\vartheta \epsilon}{\alpha_2^{k-1}} \right) \right],$$

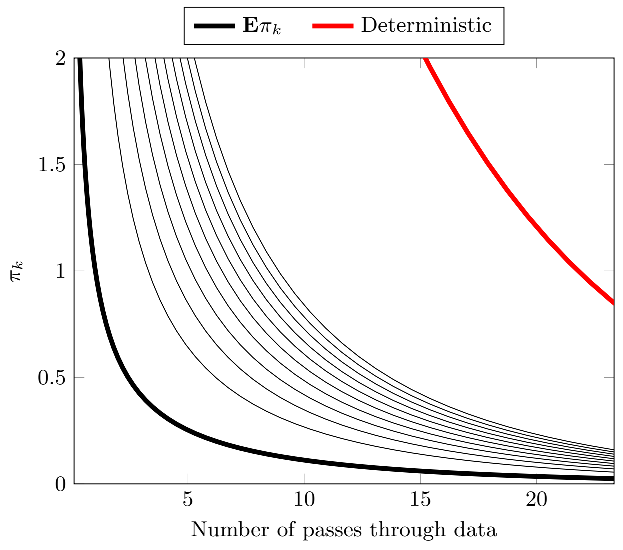


FIGURE 11.4.1. An illustration of the bounds derived in Theorem 11.4.2; this figure plots the non-asymptotic bound shown in (11.4.9). The thick black line (bottom left) shows the bound in expectation (see Part 1 of Theorem 11.4.2). For comparison, the thick red line (top right) shows the deterministic bound on the distance to the solution [FS12, Theorem 3.1]. The thin lines in between give the bounds on $\pi_k - \rho^k \pi_0$ that correspond to probabilities 10^{-i} for $i = 10, 20, \ldots, 100$. Assume $M = 300$, $\beta = 0.9$, and $\rho = 0.9$.

for some positive δ independent of ϑ and α_2. Now consider the case in which $\mathcal{O}$ depends asymptotically only on k. Take the logarithm of both sides of (11.4.9):

$$\log \Pr(\pi_k - \rho^k \pi_0 \geq \epsilon) \leq \bar{\alpha}_k \log \left(\frac{\vartheta \epsilon}{\bar{\alpha}_k n \lambda \alpha_2^{k-1}} \right) + \bar{\alpha}_k - \frac{\vartheta \epsilon}{n \lambda \alpha_2^{k-1}} = \mathcal{O}\left(-\frac{\epsilon}{\alpha_2^{k-1}} \right).$$

This implies (9.1.3). $\square$

Corollary 11.4.1 (Overwhelming tail bounds)**.** Under schedule (11.4.4), for k fixed, there exists for all $A > 0$ a constant $C_A > 0$ such that

$$\Pr\left(\pi_k - \rho^k \pi_0 \geq \epsilon\right) \leq C_A \epsilon^{-A}.$$

Take ϵ fixed. There exists a constant $C_A > 0$ such that for all $A > 0$,

$$\Pr\left(\pi_k - \rho^k \pi_0 \geq \epsilon\right) \leq C_A A^{-k}.$$

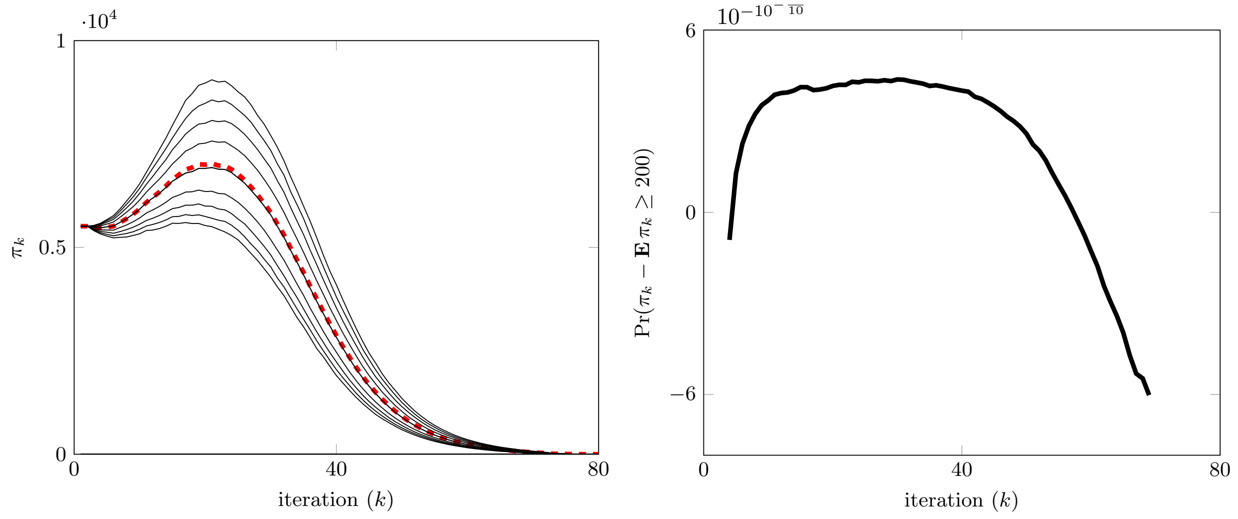


FIGURE 11.5.1. Left panel: distance to solution for quantiles $1 - 0.5^j$ and 0.5^j, $j = -5 : 5$. Right panel: probability of the deviation from expected value against a log-log y-axis, which exhibits the tail that converges with a doubly-exponential tail.

PROOF. Because the required result follows from Theorem 11.4.2, we can pick up from the proof of that result. In particular, the right-hand side of (11.4.9) can be equivalently expressed in two ways as

$$\left(\frac{\exp(1)}{\bar\alpha_k}\cdot\frac{\vartheta\epsilon}{n\lambda\alpha_2^{k-1}}\right)^{\bar\alpha_k}\exp\left(-\frac{\vartheta\epsilon}{n\lambda\alpha_2^{k-1}}\right) = \begin{cases}\mathcal{O}(1)\cdot\epsilon^{\bar\alpha_k}\exp(-\epsilon\cdot\mathcal{O}(1))\\ \exp(\phi_1(k))\exp\left(-\exp\left(\phi_2(k)\right)\right),\end{cases}$$

where

$$\phi_1(k) := \bar\alpha_k\log\left(\vartheta\epsilon\alpha_2/\bar\alpha_k\lambda\right) + \bar\alpha_k - k\bar\alpha_k\log\alpha_2 \quad\text{and}\quad \phi_2(k) := \log(\vartheta\epsilon\alpha_2/\lambda) - k\log\alpha_2,$$

and the notation $\mathcal{O}(1)$ stands for positive constants. The result then follows from Lemma B.1.1. □

11.5. Numerical experiments

Figure 11.5.1 shows the results of a Monte Carlo simulation on a logistic regression problem,

$$\underset{x}{\text{minimize}}\qquad\sum_{i=1}^{m}\log(1+\exp[-b_i a_i^T x]),$$

$a_i \in \mathbf{R}^n$ is a vector of input features, and $b_i \in \{-1, 1\}$ is the corresponding observation. For this problem, we generate a dataset with $M = 100$ pairs (a_i, b_i) of random points. Algorithm (11.0.1) where the $\epsilon_k \propto \beta^k$ with $\beta \approx .91$, is run 10K times on this fixed dataset. The starting point between runs is the same, and the only difference is the randomness of the sampling. Figure 11.5.1 summarizes the results of this experiment. As expected, the sample paths are concentrated tightly around the mean. Furthermore, the probability of deviating from the mean decays doubly-exponentially (cf.Theorem 11.4.2), as evidenced by the linear tail shown in the right panel.

Part 5

Paths forward

CHAPTER 12

Paths forward

As this thesis draws to its conclusion, we pause to reflect on possible directions of future research. The fields of proximal gradient and its stochastic variants remain a vibrant field of research. This chapter highlights some recent progress made in tackling problems of the form (1.3.1) and possible directions of future research.

12.1. Forward backward envelope

Recently a new approach to solving composite problems of the form (1.3.2) has been developed with very promising numerical results. This relies on a special kind of smoothing for problems with composite structure where f is twice differentiable, the *forward-backwards envelope*:

$$\mathbf{fenv}^{\gamma}_{f,g}(x) = f(x) - \frac{\gamma}{2}\|\nabla f(x)\|^2 + \mathbf{env}^{\gamma I}_{g}(x - \gamma\nabla f(x)).$$

When $\gamma \in (0, 1/L)$ it can be shown that any stationary point of $\mathbf{fenv}$ is also a solution of (1.3.2). And though $\mathbf{fenv}^{\gamma}_{f,g}$ is nonconvex, it is very well behaved in many ways, including a computable gradient

$$\nabla\mathbf{fenv}^{\gamma}_{f,g}(x) = \frac{1}{\gamma}(I - \gamma\nabla^2 f(x))(x - \mathbf{prox}_{\gamma g}(x - \gamma\nabla f(x)).$$

12.2. Proximal Quasi-Newton

Much of our discussion revolves around techniques for solving the Newton systems (9.4.2) that arise in the implementation of an interior method for solving QPs. The Sherman-Woodbury formula features prominently because it is a convenient vehicle for taking advantage of the structure of the Hessian approximations and the structured matrices that typically define QS functions. Other alternatives, however, may be preferable, depending on the application.

For example, we might choose to reduce the 3-by-3 matrix in (9.4.2) to an equivalent symmetrized system

$$\begin{pmatrix} -Q & A^T \\ A & D \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta s \end{pmatrix} = - \begin{pmatrix} r_d \\ r_p - V^{-1} r_\mu \end{pmatrix}$$

with $D := V^{-1}S$. As described by [BW08], Krylov-based method, such as MINRES [PS75], may be applied to a preconditioned system, using the preconditioner

$$P = \begin{pmatrix} -\mathcal{L}(u) & \\ & D \end{pmatrix}$$

where $\mathcal{L}(u)$ is defined in (9.4.5). This "ideal" preconditioner clusters the spectrum into three distinct values, so that in exact arithmetic, MINRES would converge in three iterations. The application of the preconditioner requires solving systems with $\mathcal{L}$ and D, and so all of the techniques discussed in §9.5 apply. One benefit, however, which we have not explored here, is that the preconditioning approach allows us to approximate $\mathcal{L}^{-1}(u)$, rather than to compute it exactly, which may yield computational efficiencies for some problems.

Part 6

Appendix

APPENDIX A

Appendix: Cutting plane methods

A.1. Some General Facts

Lemma A.1.1. (**Volume of a Parabolid**) Let ω_n be the volume of the n unit ball. Then

$$\mathrm{vol}(\{(t,x) \mid \tfrac{K}{2}\|x\|^2 \le t \le h\}) = \frac{2\omega_n}{n[K/2]^{n/2}} \cdot h^{(2+n)/2}.$$

Proof. This proof is an application of the "disk method" in calculus. Since

$$\frac{K}{2}\|x\|^2 \le t \iff \|x\| \le \sqrt{2t/K},$$

each slice of $\mathcal{P}$ in the first dimension is a $n-$ball of radius $\sqrt{2t/K}$, with area $\omega_n(2t/K)^{n/2}$. We integrate this from 0 to h, i.e.,

$$\mathrm{vol}(\mathcal{P}) = \omega_n \int_0^h [2t/K]^{n/2} dt = \frac{\omega_n}{[K/2]^{n/2}} \int_0^h t^{n/2} dt = \frac{2\omega_n}{n[K/2]^{n/2}} \cdot h^{(n+2)/2}.$$

□

Lemma A.1.2. (Volume approximations for MIE of the unit circular cone) Let $\mathcal{K} = \mathrm{conv}\big\{\{(1,x) \mid \frac{1}{2}\|x\|^2 \le 1\}, 0\big\}$. Then

$$\frac{\omega_{n+1}}{4^{n+1}} \le \mathrm{size}(\mathcal{K}) \le \frac{2^{n/2+1}\omega_n}{n}$$

Proof. (**Lower Bound**) We can inscribe a $n+1$ dimensional ball of radius $1/(2\sqrt{1.25}+1)$ in the unit cone. Since the MIE is larger than this,

$$\frac{\omega_{n+1}}{(2\sqrt{1.25}+1)^{n+1}} \le \mathrm{size}(\mathcal{K}).$$

(**Upper Bound**) Since $\mathrm{size}(\mathcal{K}) \le \mathrm{vol}(\mathcal{K})$, we can use Lemma A.1.1. □

Lemma A.1.3. Let ω_n be the volume of a unit sphere. Then

$$\frac{2}{n+1} \le \omega_n^{2/(2+n)} \le \frac{12}{n+1}$$

PROOF. Multiplying both sides by $n+1$, we get

$$\begin{aligned}(n+1)\omega_n^{2/(2+n)} &= (n+1)\left(\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}\right)^{2/(2+n)} \\ &\overset{(a)}{\le} (n+1)\left(\frac{\pi^{\frac{n}{2}}}{\sqrt{2\pi}n^{\frac{n+3}{2}}e^{-\frac{n+2}{2}}}\right)^{2/(2+n)} \\ &= (n+1)\frac{\pi^{\frac{n}{2+n}}}{\sqrt{2\pi}n^{\frac{n+3}{n+2}}e^{-1}} = \frac{n+1}{n^{1+\frac{1}{n+2}}}\frac{\pi^{\frac{n}{2+n}}}{\sqrt{2\pi}e^{-1}} \le \left(1+\frac{1}{n}\right)\frac{3\pi e}{\sqrt{2\pi}} \le \frac{6\pi e}{\sqrt{2\pi}} \le 12\end{aligned}$$

(a) comes from Sterling's lower bound. The lower bound is proved in an analogous way

$$\begin{aligned}(n+1)\omega_n^{2/(2+n)} &= (n+1)\left(\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}\right)^{2/(2+n)} \\ &\ge (n+1)\left(\frac{\pi^{\frac{n}{2}}}{\sqrt{2\pi}\cdot n^{n+\frac{1}{2}}e^{\frac{1}{12n}-n}}\right)^{2/(2+n)} \\ &= (n+1)\frac{\pi^{\frac{n}{2+n}}}{\sqrt{2\pi}\cdot n^{\frac{n+3}{2+n}}e^{\left(\frac{2}{2+n}\right)\left(\frac{1}{12n}-n\right)}} \ge \frac{n+1}{n}\frac{10}{\sqrt{2\pi}}\end{aligned}$$

□

APPENDIX B

Appendix: Tail bounds for proximal gradient descent

B.1. Auxiliary results

Lemma B.1.1. Suppose that

$$\phi_1(k) = \mathcal{O}(k^{\mathcal{O}(1)}) \exp(-\mathcal{O}(k^{\mathcal{O}(1)})),$$

$$\phi_2(k) = \exp(\mathcal{O}(k^{\mathcal{O}(1)})) \exp(-\exp(\mathcal{O}(k^{\mathcal{O}(1)}))),$$

where $\mathcal{O}(1)$ stands for positive constants. Then for each $A > 0$ there exists a positive constant C_A such that

$$\phi_1(k) \leq C_A k^{-A}, \tag{B.1.1}$$

$$\phi_2(k) \leq C_A A^{-k}. \tag{B.1.2}$$

Proof. The statement follows by taking the logarithms on both sides of (B.1.1) and (B.1.2). □

Lemma B.1.2. For $y \in (0, 1)$ and $x \in [0, 1]$,

$$(1 - x)^{1-1/\log y} \leq \prod_{i=0}^{\infty} (1 - xy^i). \tag{B.1.3}$$

Proof. To prove the lower bound, we use the following fact:

$$\ln(1 - x) \geq -\frac{x}{1 - x} \quad \text{for all} \quad x \in [0, 1).$$

Therefore,

$$\begin{aligned}\prod_{i=1}^{\infty}(1-xy^i) &= \exp\left(\sum_{i=1}^{\infty}\log\left(1-xy^i\right)\right)\\ &\geq \exp\left(\sum_{i=1}^{\infty}-\frac{y^i}{1/x-y^i}\right)\\ &\geq \exp\left(-\int_0^{\infty}\frac{y^i}{1/x-y^i}di\right)\\ &= \exp\left(-\frac{\log(1-x)}{\log(y)}\right)\geq (1-x)^{-1/\log y}.\end{aligned}$$

Thus,

$$\prod_{i=0}^{\infty}(1-xy^i)=(1-x)\prod_{i=1}^{\infty}(1-xy^i)\geq(1-x)^{1-1/\log y},$$

as required. □

Lemma B.1.3. For $y\in(0,1)$ and $x\in[0,1]$,

$$\exp\left(-\frac{\log(1-x/y)-\log(1-xy^{N+1})}{\log(y)}\right)\leq\prod_{i=0}^{N}(1-xy^i).$$

Proof. Similar to the proof of the previous inequality

$$\begin{aligned}\prod_{i=1}^{N}(1-xy^i) &= \exp\left(\sum_{i=1}^{N}\log\left(1-xy^i\right)\right)\\ &\geq \exp\left(\sum_{i=1}^{N}-\frac{xy^i}{1-xy^i}\right)\\ &\geq \exp\left(-\int_0^{N}\frac{xy^i}{1-xy^i}di\right)\\ &\geq \exp\left(-\frac{\log(1-x)-\log\left(1-xy^N\right)}{\log(y)}\right).\end{aligned}$$

Thus,

$$\begin{aligned}\prod_{i=0}^{N}(1-xy^i) &= \prod_{i=1}^{N+1}\left(1-(x/y)y^i\right)\\ &\geq \exp\left(-\frac{\log(1-x/y)-\log(1-xy^{N+1})}{\log(y)}\right),\end{aligned}$$

as required. □

Lemma B.1.4. Let $k > 0$, $\mu > 0$, and $\epsilon > 0$. Then for $y \in (0,1)$ and $x \in (0,1]$,

$$\inf_{\theta > 0} \left\{ \exp(-\theta \epsilon \nu) \prod_{i=0}^{N-1} \left(1 - \theta x y^i\right)^{-k} \right\} \le \left(\frac{\exp(1)}{\alpha} \cdot \frac{\epsilon \nu}{x} \right)^{\alpha} \exp\left(-\frac{\epsilon \nu}{x} \right),$$

where $\alpha = \frac{1}{k}\left(\frac{1}{\log(1/y)} + 1\right)$.

Proof. By inverting both sides of (B.1.3) we obtain the following inequality

$$\prod_{i=0}^{\infty} (1 - x y^i)^{-k} \le \exp\left(-\log(1-x) \left[\frac{1}{\log(1/y)} + 1 \right] \right). \tag{B.1.4}$$

Therefore, for $\epsilon \ge \alpha x / v$,

$$\begin{aligned}
&\inf_{\theta > 0} \left\{ \exp(-\theta \epsilon \nu) \prod_{i=0}^{N-1} (1 - \theta x y^i)^{-k} \right\} \\
&\quad \le \inf_{\theta > 0} \left\{ \exp(-\theta \epsilon \nu) \prod_{i=0}^{\infty} (1 - \theta x y^i)^{-k} \right\} \\
&\quad \overset{(i)}{\le} \inf_{\theta > 0} \left\{ \exp\left(-\frac{1}{k} \left[\frac{1}{\log(1/y)} + 1 \right] \log(1 - \theta x) - \theta v \epsilon \right) \right\} \\
&\quad = \inf_{\theta > 0} \left\{ \exp\left(-\alpha \log(1 - \theta x) - \theta \epsilon \nu \right) \right\} \\
&\quad \overset{(ii)}{=} \exp\left(-\alpha \log\left(1 - \left(\frac{1}{x} - \frac{\alpha}{v\epsilon} \right) x \right) - \left(\frac{1}{x} - \frac{\alpha}{v\epsilon} \right) v\epsilon \right) \\
&\quad = \left(\frac{\exp(1)}{\alpha} \cdot \frac{\epsilon \nu}{x} \right)^{\alpha} \exp\left(-\frac{\epsilon \nu}{x} \right),
\end{aligned}$$

where (i) follows from (B.1.4); and (ii) uses the substitution $\theta = 1/x - \alpha / v\epsilon$, which can be shown to be the optimal choice of θ. Because $\theta > 0$, $\epsilon > \alpha x / v$. □

For the remainder of this section, define the sample average to be

$$S_m := \frac{1}{m} \sum_{i}^{m} X_i$$

for a sequence of random variables $\{X_1, \dots, X_m\}$.

B.2. Sampling Bounds

Theorem B.2.1 ([Hoe63, Theorem 2])**.** Consider independent random variables $\{X_1, \dots, X_m\}$, $X_i : \Omega \to \mathbf{R}$. If the random variables are bounded, i.e.,

$$d := \sup_{\omega \in \Omega} X_i(\omega) - \inf_{\omega \in \Omega} X_i(\omega)$$

is finite, then

$$\Pr\left(S_m - \mathbf{E}S_m \geq \epsilon\right) \leq \exp\left(-2m\epsilon^2/d^2\right)$$

Theorem B.2.2 ([Ser74, Corollary 1.1])**.** Let $x_1, \dots, x_M$ be a population, $\{X_1, \dots, X_m\}$ be samples drawn without replacement from the population, and let

$$d := \max_i x_i - \min_i x_i.$$

Then

$$\Pr\left(S_m - \mathbf{E}S_m \geq \epsilon\right) \leq \exp\left(-\epsilon^2/\eta_m\right), \quad \text{where} \quad \eta_m = \frac{d^2}{2m}\left(1 - \frac{m-1}{M}\right).$$

Because η_m is strictly decreasing in m, the Serfling bound is uniformly better than the Hoeffding bound. Note that the Serfling bound is not tight: in particular, when $M = m$ (i.e., $S_m = \mathbf{E}S_m$), the bound is not zero (except for degenerate population).

APPENDIX C

Apendix: Proximal methods

C.1. QS Representation for a quadratic

Here we derive the QS representation of a support function that includes an explicit quadratic term:

$$g(x) = \sup_y \{y^T(B_0x + d_0) - \tfrac{1}{2}y^TQy \mid A_0y \succeq_{\mathcal{K}_0} b_0\}.$$

Let R be such that $R^TR = Q$. We can then write the quadratic function in the objective as a constraint on its epigraph, i.e.,

$$g(x) = \sup_{y,\,t} \{y^T(B_0x + d_0) - \tfrac{1}{2}t \mid A_0y \succeq_{\mathcal{K}} b_0,\ \|Ry\|^2 \le t\}.$$

Next we write the constraint $\|Ry\|^2 \le t$ as a second-order cone constraint:

$$\begin{aligned}
\|Ry\|^2 \le t \iff & \|Ry\|^2 \le \frac{(t+1)^2 - (t-1)^2}{4} \\
\iff & \|Ry\|^2 + \left(\frac{t-1}{2}\right)^2 \le \left(\frac{t+1}{2}\right)^2 \\
\iff & \sqrt{\|Ry\|^2 + \left(\frac{t-1}{2}\right)^2} \le \frac{t+1}{2} \\
\iff & \left\| \begin{pmatrix} 0 & \frac{1}{2} \\ R & 0 \end{pmatrix} \begin{pmatrix} y \\ t \end{pmatrix} + \begin{pmatrix} -\frac{1}{2} \\ 0 \end{pmatrix} \right\| \le \frac{t+1}{2} \\
\iff & \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ R & 0 \end{pmatrix} \begin{pmatrix} y \\ t \end{pmatrix} \preceq_Q \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 0 \end{pmatrix}.
\end{aligned}$$

Concatenating this with the original constraints gives a QS function with parameters

$$A = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ R & 0 \\ A_0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 0 \\ b_0 \end{pmatrix}, \quad d = \begin{pmatrix} d_0 \\ -\frac{1}{2} \end{pmatrix}, \quad B = \begin{pmatrix} B_0 \\ 0 \end{pmatrix}, \quad \mathcal{K} = \mathcal{Q}^{n+2} \times \mathcal{K}_0.$$

Bibliography

[AA13] E. Andersen and K. Andersen, *Mosek modeling manual*, http://mosek.com (2013).

[ABBP11] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto, *An l1-Laplace robust Kalman smoother*, IEEE Transactions on Automatic Control **56** (2011), no. 12, 2898–2911.

[ABP13] A. Y. Aravkin, J. V. Burke, and G. Pillonetto, *Sparse/robust estimation and Kalman smoothing with nonsmooth log-concave densities: Modeling, computation, and theory*, J. Mach. Learn. Res. **14** (2013), no. 1, 2689–2728.

[ACDL14] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford, *A reliable effective terascale linear learning system.*, Journal of Machine Learning Research **15** (2014), no. 1, 1111–1133.

[ADV10] M. Andersen, J. Dahl, and L. Vandenberghe, *Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones*, Math. Program. Comp. **2** (2010), no. 3-4, 167–201 (English).

[AV95] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Mathematical Programming **69** (1995), no. 1-3, 1–43.

[Azu67] K. Azuma, *Weighted sums of certain dependent random variables*, Tohoku Mathematical Journal **19** (1967), no. 3, 357–367.

[BB88] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal. **8** (1988), 141–148.

[BBE$^+$03] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song, *Dimensionality reduction via sparse support vector machines*, Journal of Machine Learning Research **3** (2003), no. Mar, 1229–1243.

[BC90] M. J. Best and N. Chakravarti, *Active set algorithms for isotonic regression; a unifying framework*, Math. Program. **47** (1990), no. 1, 425–439.

[BCNO12] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak, *A family of second-order methods for convex l1-regularized optimization*, Unpublished: Optimization Center: Northwestern University, Tech Report (2012).

[BCNW12] R. H. Byrd, G. Chin, J. Nocedal, and Y. Wu, *Sample size selection in optimization methods for machine learning*, Math. Program. **134** (2012), 127–155.

[BDEL03] S. Ben-David, N. Eiron, and P. M. Long, *On the difficulty of approximately maximizing agreements*, Journal of Computer and System Sciences **66** (2003), no. 3, 496–514.

[BEKS14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: A fresh approach to numerical computing*, November 2014, 1411.1607.

[Ber09] D. P. Bertsekas, *Convex optimization theory*, Athena Scientific Belmont, MA, 2009.

[BF12] S. Becker and J. Fadili, *A quasi-Newton proximal splitting method*, Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2012, pp. 2618–2626.

[BGT81] R. G. Bland, D. Goldfarb, and M. J. Todd, *The ellipsoid method: A survey*, Operations research **29** (1981), no. 6, 1039–1091.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 144–152.

[BGVDM94] O. Bahn, J.-L. Goffin, J.-P. Vial, and O. Du Merle, *Experimental behavior of an interior point cutting plane algorithm for convex programming: an application to geometric programming*, Discrete Applied Mathematics **49** (1994), no. 1-3, 3–23.

[BH13] J. V. Burke and T. Hoheisel, *Epi-convergent smoothing with applications to convex composite functions*, SIAM J. Optim. **23** (2013), no. 3, 1457–1479.

[Bis06] C. M. Bishop, *Pattern recognition*, Machine Learning **128** (2006), 1–58.

[BLM13] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*, Oxford University Press, 2013.

[BLS15] S. Bubeck, Y. T. Lee, and M. Singh, *A geometric alternative to Nesterov's accelerated gradient descent*, arXiv preprint arXiv:1506.08187 (2015).

[BO17] A. Basu and T. Oertel, *Centerpoints: A link between optimization and convex geometry*, SIAM Journal on Optimization **27** (2017), no. 2, 866–889.

[Bro86] L. D. Brown, *Fundamentals of statistical exponential families with applications in statistical decision theory*, Lecture Notes-monograph series **9** (1986), i–279.

[BS11] A. Barbero and S. Sra, *Fast Newton-type methods for total variation regularization.*, Intern. Conf. on Machine Learning (L. Getoor and T. Scheffer, eds.), Omnipress, 2011, pp. 313–320.

[BS14] A. Barbero and S. Sra, *Modular proximal optimization for multidimensional total-variation regularization*, 2014, arXiv 0902.0885.

[BT09] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imag. Sci. **2** (2009), no. 1, 183–202.

[Bub15] S. Bubeck, *Convex optimization: Algorithms and complexity*, Foundations and Trends in Machine Learning **8** (2015), no. 3-4, 231–357.

[BV04] D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, Journal of the ACM **51** (2004), no. 4, 540–556.

[BV07] S. Boyd and L. Vandenberghe, *Localization and cutting-plane methods*, From Stanford EE 364b lecture notes (2007).

[BVS08] S. Boyd, L. Vandenberghe, and J. Skaf, *Analytic center cutting-plane method*, 2008.

[BW08] M. Benzi and A. J. Wathen, *Some preconditioning techniques for saddle point problems*, Model order reduction: theory, research aspects and applications, Springer, 2008, pp. 195–211.

[CB02] G. Casella and R. L. Berger, *Statistical inference*, vol. 2, Duxbury Pacific Grove, CA, 2002.

[CL93] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Program. **62** (1993), no. 1-3, 261–275.

[CL06] F. Chung and L. Lu, *Concentration inequalities and martingale inequalities: a survey*, Internet Mathematics **3** (2006), no. 1, 79–127.

[CLO14] Y. Chen, G. Lan, and Y. Ouyang, *Optimal primal-dual methods for a class of saddle point problems*, SIAM Journal on Optimization **24** (2014), no. 4, 1779–1814.

[CMMP13] H. H. Chin, A. Madry, G. L. Miller, and R. Peng, *Runtime guarantees for regression problems*, Proceedings of the 4th conference on Innovations in Theoretical Computer Science, ACM, 2013, pp. 269–282.

[CP11] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, Fixed-point algorithms for inverse problems in science and engineering, Springer, 2011, pp. 185–212.

[CRT06] E. J. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math. **59** (2006), no. 8, 1207–1223.

[CV95] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning **20** (1995), no. 3, 273–297.

[DBS10] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, *Tuning support vector machines for minimax and neyman-pearson classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010), no. 10, 1888–1898.

[DES82] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal. **19** (1982), no. 2, 400–408.

[DFR16] D. Drusvyatskiy, M. Fazel, and S. Roy, *An optimal first order method based on optimal quadratic averaging*, arXiv preprint arXiv:1604.06543 (2016).

[Don06] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory **52** (2006), no. 4, 1289–1306.

[dOSL14] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal, *Convex proximal bundle methods in depth: a unified analysis for inexact oracles*, Mathematical Programming **148** (2014), no. 1-2, 241–277.

[DSSSC08] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, *Efficient projections onto the ℓ_1-ball for learning in high dimensions*, Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 272–279.

[ES10] G. Emiel and C. Sagastizábal, *Incremental-like bundle methods with application to energy planning*, Computational Optimization and Applications **46** (2010), no. 2, 305–332.

[ET99] I. Ekeland and R. Temam, *Convex analysis and variational problems*, SIAM, 1999.

[Fáb00] C. I. Fábián, *Bundle-type methods for inexact data*, Central European Journal of Operations Research **8** (2000), no. 1, 35–55.

[FCH$^+$08] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, *LIBLINEAR: A library for large linear classification*, Journal of Machine Learning Research **9** (2008), 1871–1874.

[Fen49] W. Fenchel, *On conjugate convex functions*, Canad. J. Math **1** (1949), no. 73-77.

[FG13] M. P. Friedlander and G. Goh, *Tail bounds for stochastic approximation*, arXiv preprint arXiv:1304.5586 (2013).

[FG17] ______, *Efficient evaluation of scaled proximal operators*, Electronic Transactions on Numerical Analysis **46** (2017), 1–22.

[FGRW12] V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu, *Agnostic learning of monomials by halfspaces is hard*, SIAM Journal on Computing **41** (2012), no. 6, 1558–1590.

[FHT10] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*, Journal of statistical software **33** (2010), no. 1, 1.

[FS95] Y. Freund and R. E. Schapire, *A desicion-theoretic generalization of on-line learning and an application to boosting*, European conference on computational learning theory, Springer, 1995, pp. 23–37.

[FS12] M. P. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM Journal on Scientific Computing **34** (2012), no. 3, A1380–A1405.

[GCGF16] G. Goh, A. Cotter, M. Gupta, and M. P. Friedlander, *Satisfying real-world goals with dataset constraints*, Advances in Neural Information Processing Systems, 2016, pp. 2415–2423.

[GGBHD04] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, *Result analysis of the NIPS 2003 feature selection challenge*, Advances Neural Inform. Processing Systems **17** (2004), 545–552.

[GKT51] D. Gale, H. W. Kuhn, and A. W. Tucker, *Linear programming and the theory of games*, Activity analysis of production and allocation **13** (1951), 317–335.

[GL89] G. H. Golub and C. F. V. Loan, *Matrix computations*, second ed., Johns Hopkins University Press, Baltimore, 1989.

[GLS12] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, vol. 2, Springer Science & Business Media, 2012.

[Grü60] B. Grünbaum, *Partitions of mass-distributions and of convex bodies by hyperplanes*, Pacific Journal of Mathematics **10** (1960), no. 4, 1257–1261.

[GV99] J.-L. Goffin and J.-P. Vial, *A two-cut approach in the analytic center cutting plane method*, Mathematical methods of Operations Research **49** (1999), no. 1, 149–169.

[H$^+$64] P. J. Huber et al., *Robust estimation of a location parameter*, The Annals of Mathematical Statistics **35** (1964), no. 1, 73–101.

[HL00] D. Hosmer and S. Lemeshow, *Applied logistic regression*, Wiley-Interscience, 2000.

[Hoe63] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. American Stat. Assoc. **58** (1963), no. 301, 13–30.

[HPS$^+$16] M. Hardt, E. Price, N. Srebro, et al., *Equality of opportunity in supervised learning*, Advances in Neural Information Processing Systems, 2016, pp. 3315–3323.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning. data mining, inference, and prediction*, Springer, 2001.

[Hub11] P. J. Huber, *Robust statistics*, Springer, 2011.

[Imh61] J. P. Imhof, *Computing the distribution of quadratic forms in normal variables*, Biometrika (1961), 419–426.

[JAB11] R. Jenatton, J.-Y. Audibert, and F. Bach, *Structured variable selection with sparsity-inducing norms*, Journal of Machine Learning Research **12** (2011), no. Oct, 2777–2824.

[JMBO10] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski, *Proximal methods for sparse hierarchical dictionary learning*, Proc. 27th Intern. Confer. Machine Learning (ICML-10), 2010, pp. 487–494.

[Joa99] T. Joachims, *Svmlight: Support vector machine*, SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund **19** (1999), no. 4.

[KBJ78] R. Koenker and G. Bassett Jr, *Regression quantiles*, Econometrica: journal of the Econometric Society (1978), 33–50.

[Kel60] J. E. Kelley, Jr, *The cutting-plane method for solving convex programs*, Journal of the society for Industrial and Applied Mathematics **8** (1960), no. 4, 703–712.

[Kha79] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Doklady Akademiia Nauk SSSR, vol. 244, 1979, pp. 1093–1096.

[KKL$^+$07] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale L1-regularized least squares*, IEEE J. Sel. Top. Signal Process. **1** (2007), no. 4, 606–617.

[Kle08] A. Klenke, *Probability theory: a comprehensive course*, Springer Verlag, London, 2008.

[Koe05] R. Koenker, *Quantile regression*, no. 38, Cambridge university press, 2005.

[KT51] H. W. Kuhn and A. W. Tucker, *Nonlinear programming*, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (1951), 481492.

[KV14] S. Karimi and S. Vavasis, *IMRO: a proximal quasi-Newton method for solving l_1-regularized least squares problem*, 2014, arXiv:1401.4220.

[Lev65] A. Y. Levin, *On an algorithm for the minimization of convex functions*, Soviet Mathematics Doklady, vol. 160, 1965, pp. 1244–1247.

[LHU96] C. Lemaréchal and J. Hiriart-Urruty, *Convex analysis and minimization algorithms i*, Grundlehren der mathematischen Wissenschaften **305** (1996).

[Lic13] M. Lichman, *UCI machine learning repository*, 2013.

[LM01] Y.-J. Lee and O. L. Mangasarian, *Ssvm: A smooth support vector machine for classification*, Computational optimization and Applications **20** (2001), no. 1, 5–22.

[LNN95] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, *New variants of bundle methods*, Math. Program. **69** (1995), 111–147.

[Lor13] D. A. Lorenz, *Constructing test instances for basis pursuit denoising*, IEEE Trans. Sig. Proc. **61** (2013), no. 5, 1210–1214.

[LSB81] C. Lemarechal, J.-J. Strodiot, and A. Bihain, *On a bundle algorithm for nonsmooth optimization*, Nonlinear programming **4** (1981), no. 0.

[LSS14] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim. **24** (2014), no. 3, 1420–1443.

[LT93a] Z. Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: a general approach*, Annals of Operations Research **46** (1993).

[LT93b] Z. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res. **46** (1993), no. 1, 157–178.

[LTTT14] R. Lockhart, J. Taylor, R. J. Tibshirani, and R. Tibshirani, *A significance test for the lasso*, Annals of statistics **42** (2014), no. 2, 413.

[LV03] L. Lovász and S. Vempala, *Hit-and-run is fast and fun*, preprint, Microsoft Research (2003).

[LZOX14] Y. Lou, T. Zeng, S. Osher, and J. Xin, *A weighted difference of anisotropic and isotropic total variation model for image processing*, Tech. report, Department of Mathematics, UCLA, 2014.

[Mag74] T. L. Magnanti, *Fenchel and Lagrange duality are equivalent*, Mathematical Programming **7** (1974), no. 1, 253–258.

[MAT10] MATLAB, *version 7.10.0 (r2010a)*, The MathWorks Inc., Natick, Massachusetts, 2010.

[McC84] P. McCullagh, *Generalized linear models*, European Journal of Operational Research **16** (1984), no. 3, 285–292.

[Meh00] S. Mehrotra, *Volumetric center method for stochastic convex programs using sampling*, (2000).

[Mie12] K. Miettinen, *Nonlinear multiobjective optimization*, vol. 12, Springer Science & Business Media, 2012.

[MM07] G. S. Mann and A. McCallum, *Simple, robust, scalable semi-supervised learning via expectation regularization*, Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 593–600.

[NB00] A. Nedic and D. Bertsekas, *Convergence rate of incremental subgradient algorithms*, Stochastic Optimization: Algorithms and Applications (2000), 263–304.

[Nem94] A. Nemirovski, *Efficient methods in convex programming*, Lecture notes (1994).

[New65] D. J. Newman, *Location of the maximum on unimodal surfaces*, Journal of the ACM (JACM) **12** (1965), no. 3, 395–398.

[NJLS09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim. **19** (2009), no. 4, 1574–1609.

[NV08] Y. Nesterov and J.-P. Vial, *Confidence level solutions for stochastic programming*, Automatica **44** (2008), no. 6, 1559–1568.

[NW99] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, New York, 1999.

[Pan87] J.-S. Pang, *A posteriori error bounds for the linearly constrained variational inequality problem*, Math. Oper. Res. **12** (1987).

[Pas16] `http://largescale.ml.tu-berlin.de/instructions`, 2016, Accessed: 2016-11-22.

[Pla98] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, (1998).

[PS75] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, siamnumanal **12** (1975), 617–629.

[R$^+$66] R. T. Rockafellar et al., *Extension of fenchelduality theorem for convex functions*, Duke mathematical journal **33** (1966), no. 1, 81–89.

[Rad07] L. A. Rademacher, *Approximating the centroid is hard*, Proceedings of the twenty-third annual symposium on Computational geometry, ACM, 2007, pp. 302–305.

[RF10] H. L. Royden and P. M. Fitzpatrick, *Real analysis*, Prentice Hall, Boston 2010, 4th edition.

[RM51] H. Robbins and S. Monro, *A stochastic approximation method*, The Annals of Mathematical Statistics (1951), 400–407.

[Roc64] R. Rockafellar, *Duality theorems for convex functions*, Bulletin of the American Mathematical Society **70** (1964), no. 1, 189–192.

[Roc67] R. Rockafellar, *Duality and stability in extremum problems involving convex functions*, Pacific Journal of Mathematics **21** (1967), no. 1, 167–187.

[Roc70] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, 1970.

[Roc74] R. T. Rockafellar, *Conjugate duality and optimization*, SIAM, 1974.

[RW98] R. T. Rockafellar and R. J. B. Wets, *Variational analysis*, vol. 317, Springer, 1998, Corrected 3rd printing.

[Ser74] R. Serfling, *Probability inequalities for the sum in sampling without replacement*, Ann. Statist. **2** (1974), no. 1, 39–48.

[Sho12] N. Z. Shor, *Minimization methods for non-differentiable functions*, vol. 3, Springer Science & Business Media, 2012.

[SN05] C. Scott and R. Nowak, *A neyman-pearson approach to statistical learning*, IEEE Transactions on Information Theory **51** (2005), no. 11, 3806–3819.

[So13] A. M.-C. So, *Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity*, `http://www.se.cuhk.edu.hk/~manchoso/papers/inexact_GM_conv.pdf`, August 2013.

[SP95] P. B. Stark and R. L. Parker, *Bounded-variable least-squares: an algorithm and applications*, Computational Statistics **10** (1995), 129–129.

[SRB11] M. Schmidt, N. L. Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, arXiv preprint arXiv:1109.2415 (2011).

[ST16] K. Scheinberg and X. Tang, *Practical inexact proximal quasi-Newton method with global complexity analysis*, Math. Program. **160** (2016), no. 1, 495–529.

[SvdBFM09] M. Schmidt, E. van den Berg, M. P. Friedlander, and K. Murphy, *Optimizing costly functions with simple constraints: a limited-memory projected quasi-Newton algorithm*, Proc. 12th Inter. Conf. Artificial Intelligence and Stat., April 2009, pp. 448–455.

[Swa09] J. Swanson, *Conditional expectation*, Unpublished lecture notes, `http://www.swansonsite.com/W/instructional/condexp.pdf`, April 2009.

[Tar88] E. Tarasov, Khachiyan, *The method of inscribed ellipsoids*, Soviet Mathematics Doklady **298** (1988), no. 5, 1081–1085.

[TY09] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program. **117** (2009).

[Vai89] P. M. Vaidya, *A new algorithm for minimizing convex functions over convex sets*, 30th Annual Symposium on Foundations of Computer Science, IEEE, 1989, pp. 338–343.

[Van10] L. Vandenberghe, *The CVXOPT linear and quadratic cone program solvers*, 2010.

[VDBF08] E. Van Den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM Journal on Scientific Computing **31** (2008), no. 2, 890–912.

[VK82] V. N. Vapnik and S. Kotz, *Estimation of dependences based on empirical data*, vol. 40, Springer-Verlag New York, 1982.

[VV98] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 1, Wiley New York, 1998.

[Wil91] D. Williams, *Probability with martingales*, 8th ed., Cambridge University Press, England, 1991.

[WL14] P.-W. Wang and C.-J. Lin, *Iteration complexity of feasible descent methods for convex optimization.*, Journal of Machine Learning Research **15** (2014), no. 1, 1523–1548.

[WNF07] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, Tech. report, Computer Sciences Department, University of Wisconsin, Madison, October 2007.

[WNF09] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Sig. Proc. **57** (2009), no. 7, 2479–2493.

[Ye96] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Mathematical Programming **78** (1996), no. 1, 85–104.

[YHL12] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, *An improved glmnet for l1-regularized logistic regression*, Journal of Machine Learning Research **13** (2012), no. Jun, 1999–2030.

[Yip86] E. Yip, *A note on the stability of solving a rank-p modification of a linear system by the Sherman-Morrison-Woodbury formula*, SIAM J. Sci. Stat. Comput. **7** (1986), no. 2, 507–513.

[YL06] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. Royal Stat. Soc. B. **68** (2006).

[ZRY09] P. Zhao, G. Rocha, and B. Yu, *The composite absolute penalties family for grouped and hierarchical variable selection*, The Annals of Statistics (2009), 3468–3497.