Sequential Convex Programming

John Duchi (with help from Stephen Boyd and Jacob Mattingley)
Notes for EE364b, Stanford University

Spring 2018

Contents

1 Methods for nonconvex optimization problems

The power of modeling problems as convex optimization problems should by now be fairly obvious to you: convex optimization problems are (roughly) always globally solvable, and it is (roughly) always fast to solve them. There are of course caveats as problem sizes grow large, but in general, we have both desiderata: problems are solvable, and we can solve them quickly. Some problems, however, are challenging to model as convex optimization problems, at least in a global sense, and so we have to give up one of these two desiderata. With that in mind, we can consider two families of methods:

local optimization methods These methods are fast, but do not necessarily find globally optimal solutions. Even if they *do* find a globally optimal solution, it is often impossible to certify that it is indeed globally optimal.

global optimization methods These methods find global solutions and certify them (for example, branch and bound methods) but they are not always fast. (Indeed, they are often slow.)

In these notes, we investigate the first family of methods: local optimization methods. These methods are *heuristic*, they often fail to find optimal (or even feasible) points, and their results depend on their initial starting points. With that said, they are often effective in practice, and one can always run each method multiple times from multiple starting points in order to get a good enough solution.

1.1 Sequential convex programming (SCP)

Sequential convex programming (SCP) is a local optimization method for nonconvex problems that leverages convex optimization. The basic idea is simple: we handle the convex portions of the problem exactly and efficiently, while for the nonconvex portions of the problem, we model them by convex functions that are (at least locally) accurate.

One way to motivate SCP strategies is to reconsider the classical gradient and Newton methods. A typical optimization scheme for minimizing a function f iterates in the following way: at iteration k, we form a model $\widehat{f}$ of f that is "good enough" near the current iterate $x^{(k)}$, minimize that model or a regularized version of it, and repeat. To see that gradient descent is like this, note that we can rewrite the iteration

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$$

as

$$x^{(k+1)} = \operatorname*{argmin}_x \left\{ f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2\alpha_k} \left\| x - x^{(k)} \right\|_2^2 \right\}.$$

Thus, in this case, we have the first-order model $\widehat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)})$, and we regularize (to keep the points close enough that the model is accurate) by $\frac{1}{2\alpha_k} \|\cdot - x^{(k)}\|_2^2$. Newton's method, on the other hand, uses the quadratic model

$$\widehat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T \nabla^2 f(x^{(k)})(x - x^{(k)}).$$

Sequential convex programming, broadly, simply refers to using a convex model $\widehat{f}$ and repeatedly minimizing it.

To set the stage, we consider the (potentially nonconvex) problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{array} \tag{1}$$

in the variable $x \in \mathbf{R}^n$. Here, the functions f_0 and f_i are (possibly) nonconvex, and the functions h_j may be non-affine. Then the basic idea of SCP is to iterate by maintaining an estimate of the solution $x^{(k)}$ and a convex *trust region*, denoted $\mathcal{T}^{(k)} \subset \mathbf{R}^n$, over which we "trust" our solutions and models. The generic SCP strategy then forms a

- *convex* approximation $\widehat{f}_i$ of the functions f_i over the trust region $\mathcal{T}^{(k)}$
- *affine* approximation $\widehat{h}_i$ of the functions h_i over the trust region $\mathcal{T}^{(k)}$.

We then iterate by setting $x^{(k+1)}$ to be the optimal point for the approximating convex model of the original problem (1),

$$\begin{array}{ll} \text{minimize} & \widehat{f}_0(x) \\ \text{subject to} & \widehat{f}_i(x) \le 0, \quad i = 1, \ldots, m \\ & \widehat{h}_j(x) = 0, \quad j = 1, \ldots, p \\ & x \in \mathcal{T}^{(k)}. \end{array} \tag{2}$$

In the remainder of these notes, we describe several of the standard approaches to modeling problem (1) and solving (2).

2 Trust region methods

Trust region methods are the classical workhorse for sequential convex programming, and typically involve *sequential quadratic* programming. For these, we take either first- or second-order models in the approximation (2), and the trust region is typically either an ℓ_2-norm ball

$$\mathcal{T}^{(k)} = \left\{x \in \mathbf{R}^n \mid \left\|x - x^{(k)}\right\|_2 \le \rho\right\}$$

or a box

$$\mathcal{T}^{(k)} = \left\{x \in \mathbf{R}^n \mid |x_i - x_i^{(k)}| \le \rho_i, \ i = 1, \ldots, n\right\}.$$

As we will see, the former case allows somewhat more flexibility in our modeling strategies, while for the latter, we can leave indices x_i only involved in convex objectives and inequalities and linear equalities unconstrained (i.e. $\rho_i = +\infty$).

Then, for the models $\widehat{f}$ we take either an affine (first-order) Taylor approximation

$$\widehat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)})$$

or the convex part of the second order Taylor expansion,

$$\widehat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T P (x - x^{(k)}),$$

where $P = \left[\nabla^2 f(x^{(k)})\right]_+$ is the positive semidefinite part of the Hessian. That is, if $U \Lambda U^T = \nabla^2 f(x^{(k)})$ is the spectral decomposition of $\nabla^2 f(x^{(k)})$, then $P = U \left[\Lambda\right]_+ U^T$, which simply zeroes out all negative eigenvalues of $\nabla^2 f(x^{(k)})$. These approximations are good locally, but the radius ρ in the trust regions is still important so that they are good local estimates.

2.1 A numerical example

To get a sense of some of the issues in these problems, we begin by working out a small scale numerical example of a non-convex quadratic problem, minimized over a box. The problem is

$$\begin{array}{ll} \text{minimize} & f(x) = \frac{1}{2} x^T P x + q^T x \\ \text{subject to} & \|x\|_\infty \leq 1, \end{array}$$

where P is symmetric but not positive semindefinite. In this case, we may write the first-order Taylor approximation to f at a point $x^{(k)}$ as $f(x^{(k)}) + (Px^{(k)} + q)^T (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T P (x - x^{(k)})$. Then applying the idea of making a "good enough" convex approximation, we use

$$\widehat{f}(x) = f(x^{(k)}) + (Px^{(k)} + q)^T (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T P_+ (x - x^{(k)})$$

where P_+ is the projection of P onto the space of PSD matrices. In Figure 1, we display the convergence of 10 different random initializes $x^{(0)}$ in the box $\|x\|_\infty \leq 1$, in dimension $n = 20$. Here, we iteratively minimize the approximations $\widehat{f}(x)$ centered at $x^{(k)}$ over the trust regions $\mathcal{T}^{(k)} = \{x \in \mathbf{R}^n \mid \left\|x - x^{(k)}\right\|_\infty \leq \rho\}$, where $\rho = .2$.

From the figure, we see a few different results: first, the intialization changes performance, sometimes substantially. Second, it is never clear that we have actually solved the problem; none of the methods converges to the lower bound, but the lower bound may be loose. To derive *a* lower bound on the problem, we can take its Lagrange dual. In this case, the representation of the constraints is quite important; if we do not use an appropriate functional form, then the dual problem can easily be identically $-\infty$. With this in mind, we rewrite the constraint $\|x\|_\infty \leq 1$ as $x_i^2 \leq 1$ for $i = 1, \ldots, n$, obtaining Lagrangian

$$\begin{aligned} L(x, \lambda) &= \frac{1}{2} x^T P x + q^T x + \sum_{i=1}^n \lambda_i (x_i^2 - 1) \\ &= \frac{1}{2} x^T (P + 2\,\mathrm{diag}(\lambda)) x + q^T x - \mathbf{1}^T \lambda, \end{aligned}$$

so that the dual function has form

$$g(\lambda) = -\frac{1}{2} q^T (P + 2\,\mathrm{diag}(\lambda))^{-1} q - \mathbf{1}^T \lambda \quad \text{when } P + 2\,\mathrm{diag}(\lambda) \succ 0,$$

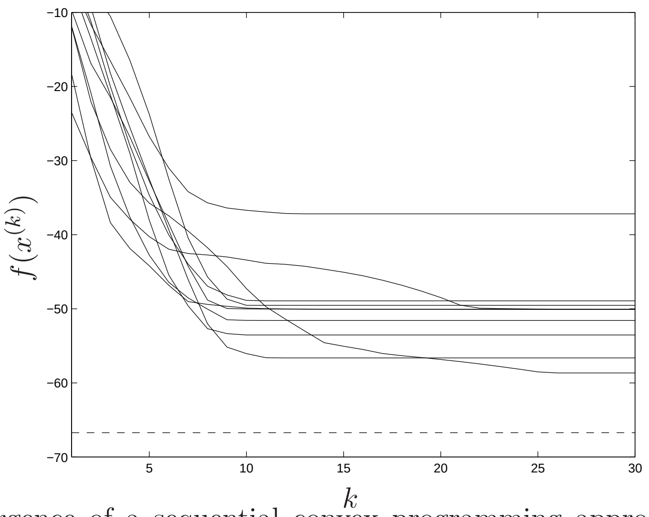


Figure 1. Convergence of a sequential convex programming approach for a nonconvex quadratic program over the box $\|x\|_\infty \leq 1$. The dashed line indicates a lower bound from a Lagrangian dual problem.

where we have been a bit fast and loose in the inversion (assuming $Pq \neq 0$ makes this rigorous). Thus, we obtain dual problem

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}q^T(P + 2\,\text{diag}(\lambda))^{-1}q - \mathbf{1}^T\lambda \\ \text{subject to} & \lambda \succeq 0,\ P + 2\,\text{diag}(\lambda) \succ 0, \end{array}$$

in variables $\lambda \succeq 0$, which is a convex optimization problem. While there may be other possible dual problems—which is common in non-convex optimization—solving this problem provides the lower bound in Fig. 1.

2.2 The classical trust region method

In the classical trust region problem, as treated (for example) by Conn, Gould, and Toint, one uses the set $\mathcal{T}^{(k)} = \{x \in \mathbf{R}^n \mid \left\|x - x^{(k)}\right\|_2\} \leq \rho$ as the trust region. Without loss of generality taking $\rho = 1$, in this case, assuming there are no constraints on x, it is not necessary to project the Hessian into the positive definite cone, because the problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^TAx + b^Tx \\ \text{subject to} & \|x\|_2 \leq 1 \end{array} \tag{3}$$

is efficiently solvable. The reasons for this involve the S-procedure and theorems of alternatives for non-convex quadratic problems (see Appendix B of [BV04]), though we can provide a reasonably simple characterization of solutions to the problem (3). Indeed, we have the following theorem.

Theorem 1. *A point $x^\star$ is optimal for the trust region problem (3) if and only if there exists $\lambda^\star \ge 0$ such that*

$$A + \lambda^\star I \succeq 0, \quad (A + \lambda^\star I)x^\star + b = 0, \quad \lambda^\star(\|x^\star\|_2 - 1) = 0.$$

If $A + \lambda^\star \succ 0$, then the solution $x^\star$ is unique.

Proof We ignore the uniqueness result, proving the rest of the theorem. First, let us suppose that the pair $(x^\star, \lambda^\star)$ satisfies the three conditions of the theorem. Writing the Lagrangian of the objective, we have

$$\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Ax + b^T x + \frac{\lambda}{2}(\|x\|_2^2 - 1).$$

Then under the three conditions, we see that

$$\mathcal{L}(x^\star, \lambda) \le \mathcal{L}(x^\star, \lambda^\star) \le \mathcal{L}(x, \lambda^\star)$$

for all $x \in \mathbf{R}^n$ and $\lambda \ge 0$, so that the pair $(x^\star, \lambda^\star)$ form a saddle point for the Lagrangian. They are thus optimal.

The question of the existence of $\lambda^\star \ge 0$ is all that remains. Let $A = UDU^T$ be the spectral decomposition of A, with $D = \operatorname{diag}(d)$ for some $d \in \mathbf{R}^n$. We shall assume that $d \not\succeq 0$, as otherwise the problem is convex and the result is immediate. Thus, let $d_1 \ge \ldots \ge d_n$ with $d_n < 0$. We consider two cases, based on whether $u_n^T b \neq 0$ for the nth eigenvector of A. In the first case, we suppose that $b^T u_n \neq 0$. Then we have that $\lim_{\lambda \to \infty} \|(A + \lambda I)^{-1} b\|_2 = 0$ and $\lim_{\lambda \downarrow -d_n} \|(A + \lambda I)^{-1} b\|_2 = \infty$, so there exists a $\lambda > -d_n > 0$ such that

$$x = -(A + \lambda I)^{-1} b \;\; \text{satisfies} \;\; \|x\|_2 = 1,$$

so that all the conditions of the theorem are satisfied.

In the so-called "hard case," which corresponds to $u_n^T b = 0$, we take $\lambda = -d_n$. Then $x_\lambda = -(A + \lambda I)^\dagger b$ has $u_n^T x_\lambda = 0$ and $A + \lambda I \succeq 0$ by assumption. If $\|x_\lambda\|_2 > 1$, then we have $\lim_{\lambda \to \infty} \|(A + \lambda I)^{-1} b\|_2 = 0$ and continuity again implies the existence of $\lambda > -d_n$ such that $x = -(A + \lambda I)^{-1} b$ with $\|x\|_2 = 1$, satisfying all conditions of the theorem. The final case that $\|x_\lambda\|_2 \le 1$, where $\lambda = -d_n$, is relatively simple as well: we set

$$x = x_\lambda + (1 - \|x_\lambda\|_2^2)^{1/2} u_n,$$

which satisfies all the conditions of the theorem: we have $\|x\|_2 = 1$ because $u_n \perp x_\lambda$ and $(A + \lambda I)u_n = 0$. □

The proof of Theorem 1 also suggests a method for solving the trust region update: we check the cases for existence of λ^*, and perform a binary search over feasible values of λ until we have satisfied the conditions of the theorem. Alternatively, there are numerous sophisticated Newton-type root-finding strategies to obtain the optimal value $\lambda^\star$ in the theorem; the book [CGT00] contains procedures and references (see also [NW06]).

2.3 Regularized methods

A variant of the trust region method is to regularize the problem instead of directly constraining the iterates to lie close to one another. Broadly, in this case, at each iteration we define some type of *model* of the function f, centered at the current iterate $x^{(k)}$, and regularize the model so that it remains accurate near $x^{(k)}$, or provides an upper bound on f itself. We typically write this model as

$$f_x(y) \approx f(y),$$

where $f_x(x) = f(x)$ so that x denotes its "centering" point. In this notation, the first-order model is

$$f_x(y) = f(x) + \nabla f(x)^T (y - x),$$

while the second order (quadratic) model is

$$f_x(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 (y - x).$$

Then instead of iterating by setting $x^{(k+1)} = \text{argmin}_x f_{x^{(k)}}(x)$, we add a regularization term $r : \mathbf{R}^n \to \mathbf{R}_+$, where $r(0) = 0$, and update

$$x^{(k+1)} = \underset{x}{\text{argmin}} \left\{ f_{x^{(k)}}(x) + r(x - x^{(k)}) \right\}. \tag{4}$$

One recent and powerful strategy, strongly related to the trust region methods, is termed *cubic* regularization of Newton's method [Gri81, NP06]. In this case, we assume the problem is unconstrained and use $r(x) = \frac{\rho}{3} \|x\|_2^3$, term, updating

$$x^{(k+1)} = \underset{x}{\text{argmin}} \left\{ f_{x^{(k)}}(x) + \frac{\rho}{3} \left\| x - x^{(k)} \right\|_2^3 \right\},$$

where f_x is the quadratic model. This cubic problem is a Lagrangian for the trust region problem with region $\mathcal{T}^{(k)} = \{x \mid \|x - x^{(k)}\|_2 \leq r\}$, so such an idea should come as relatively little surprise.

The advantage of regularized formulations (4) is that, with judicious choice of regularizer r, we can often treat them as *majorization-minimization* algorithms, that is, algorithms that sequentially minimize upper bounds on the function f that are tight at the current iterate $x^{(k)}$. For any such algorithm, we always decrease the objective, so the function values $f(x^{(k)})$ either converge or decrease to $-\infty$. For a more precise convergence rate in a reasonably general case, see Section 5 to follow.

In the case of the cubic-regularized Newton's method, we can give conditions under which it is guaranteed to be an upper bound. Indeed, assume that $\nabla^2 f(x)$ is L-Lipschitz continuous, meaning that $\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\|_{\text{op}} \leq L \left\|x - y\right\|_2$ for $x, y \in \mathbf{R}^n$ and $\left\|\cdot\right\|_{\text{op}}$ denotes the typical ℓ_2-operator norm. Then a calculation with Taylor's theorem [NP06] implies that if $f_x(y)$ is second-order expansion of f at x, then

$$|f(y) - f_x(y)| \leq \frac{L}{6} \left\|x - y\right\|_2^3,$$

so that we may take $\rho = L/2$ in the cubic problem. The solution of the resulting problem, that is, to minimize

$$\frac{1}{2}x^T Ax + b^T x + \frac{\rho}{3}\left\|x\right\|_2^3, \tag{5}$$

has similar structure to the trust region update (Theorem 1). An entirely parallel argument shows that $x^\star$ is a solution to the problem (5) if and only if

$$A + \rho I \left\|x^\star\right\|_2 \succeq 0, \quad (A + \rho I \left\|x^\star\right\|_2)x^\star + b = 0,$$

and the solution is unique if $A + \rho I \left\|x^\star\right\|_2 \succ 0$. Again, a root finding search based on taking an eigen-decomposition and finding $\left\|x^\star\right\|_2$ allows one to solve the problem.

2.4 Convex-concave procedure

TODO

2.5 A basic sequential convex programming method

There are a number of issues in actually implementing the general SCP method (2). While each step is, in principle, easy to solve (as it is a convex problem), numerous issues arise because of infeasibility, how to decide when to accept a step, and trading between feasibility of constraints and quality of the objective.

With this in mind, a typical approach is to assign *penalties* to constraint violations rather than to directly enforce the constraints, which may be impossible anyway. For a penalty method, we replace the original problem (1) with an approximation

$$\phi(x) := f_0(x) + \lambda\left(\sum_{i=1}^m \left[f_i(x)\right]_+ + \sum_{j=1}^p \left|h_j(x)\right|\right),$$

where $\lambda > 0$ is a penalty parameter to be chosen. This approximation is known as an *exact penalty* approach, because (for large enough λ) it typically does not introduce any spurious local minima, and is minimized at points where the constraints are exactly satisfied. (One can say much more about this formulation, and we will revisit it and related ideas later.) Suffice it to say, the penalization approach above can often allow progress that would otherwise be impossible, as it allows *violation* of the constraints and a more careful trading between the objective f_0, constraint violoations on f_i, and constraint violations on the h_j.

At a given iteration k in which we solve the update (2), we may compare the solution $\widetilde{x}$ and its expected objective value to the actual objective, performing a backtracking line search. Indeed, let

$$\widehat{\phi}(x) := \widehat{f}_0(x) + \lambda\left(\sum_{i=1}^m \left[\widehat{f}_i(x)\right]_+ + \sum_{j=1}^p \left|\widehat{h}_j(x)\right|\right), \tag{6}$$

where $\widehat{f}$ and $\widehat{h}$ are the convex and linear approximations to f and h, respectively. In this case, the problem (6) is convex in x, and (typically) is efficiently solvable. There are two natural approaches: one is based on decreasing and increasing the radius of the trust region, while the other performs a type of backtracking line search on the objectives.

In the former case, where we update the trust region, let us assume the trust region is $\mathcal{T}^{(k)} = \{x \in \mathbf{R}^n : \|x - x^{(k)}\| \le \rho^{(k)}\}$. Let $\alpha \in (0, \frac{1}{2})$ and $\beta^{\rm succ} > 1$ and $\beta^{\rm fail} < 1$. Then we set $\widetilde{x}$ to minimize the approximation (6) over $x \in \mathcal{T}^{(k)}$. We have *predicted decrease*

$$\widehat{\delta} = \phi(x^{(k)}) - \widehat{\phi}(\widetilde{x}),$$

and the *actual decrease*

$$\delta = \phi(x^{(k)}) - \phi(\widetilde{x}).$$

If $\delta \ge \alpha\widehat{\delta}$, meaning that we have sufficient decrease, then we accept the step and set $x^{(k+1)} = \widetilde{x}$, enlarging the trust region by $\rho^{(k+1)} = \beta^{\rm succ}\rho^{(k)}$. Otherwise, we reject the step, setting $\rho^{(k+1)} = \beta^{\rm fail}\rho^{(k)}$, and re-solving the problem from $x^{(k)}$. This is similar to the classical Armijo or backtracking line searchers, where we accept a step when the actual decrease is more than a fraction α of the predicted amount, increasing the stepsize in that case.

The second and somewhat simpler approach is a basic backtracking line search. In this case, we either solve problem (2) or (6) to obtain $\widetilde{x}$, then set $\Delta = \widetilde{x} - x^{(k)}$. Then for some $\alpha \in (0, \frac{1}{2})$ and $\beta > 1$, we backtrack beginning from $t = 1$: until

$$\phi(x^{(k)} + t\Delta) \le \phi(x^{(k)}) - \alpha t(\phi(x^{(k)}) - \widehat{\phi}(\widetilde{x})),$$

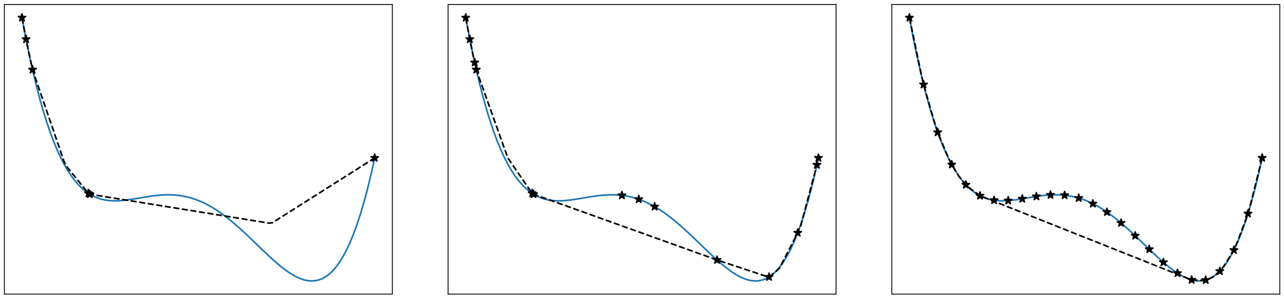
we update $t := t/\beta$. We then accept the step and update $x^{(k+1)} = x^{(k)} + t\Delta$. So long as the approximations $\widehat{f}$ and $\widehat{h}$ are locally accurate enough (e.g. if the functions f and h are differentiable), this procedure will terminate eventually.

3 Convex modeling approaches

The approaches we have described thus far are often effective. There are somewhat more sophisticated approaches that allow for even non-differentiable, nonconvex problems. In some cases, these methods encompass a more natural way to approach the problems, allowing a type of disciplined nonconvex programming.

3.1 Particle methods

Particle methods take an approach to modeling the general nonconvex problem (1) by a convex function without assuming essentially anything about the original problem. They iterate by choosing points $z_1, \ldots, z_K \in \mathcal{T}^{(k)}$, the kth trust region, and evaluating the function values $y_i = f(z_i)$. These points z_i may be chosen in many ways: uniformly at random, using quasi-Monte-Carlo methods, or at the extreme points of $\mathcal{T}^{(k)}$. The idea is to then fit the data (z_i, y_i) with the "best" convex approximation to the data, using convex approximation approaches. The advantages of these methods include that they allow for general



(a) (b) (c)

Figure 2. The particle method. (a) Four randomly sampled points z_i (plus boundaries) and resulting fitted function. (b) Twelve randomly sampled points z_i (plus boundaries) and resulting fitted function. (c) Twenty four uniformly spaced points z_i and resulting fitted function.

nondifferentiable functions, or functions for which evaluating derivatives is very challenging, and they give regional models that are accurate in a neighborhood of the current point $x^{(k)}$. Assuming enough coverage in the evaluated points $z_1, \ldots, z_K$, they can be accurate across the entire trust region $\mathcal{T}^{(k)}$. On the other hand, they can often have extraordinary sampling requirements—there exist functions for which the sample size K at each iteration must scale exponentially with the dimension.

Let us describe two fitting strategies for such problems. A simple idea is to find the tightest convex lower bound on the sampled function values $(z_i, y_i) = (z_i, f(z_i))$; in effect, this is (approximately) taking the biconjugate of f over the region $\mathcal{T}^{(k)}$. We begin by fitting f with a piecewise affine function (as f^{**} is of course the supremum of all affine functions underestimating f). Let us call the fitted function $h : \mathbf{R}^n \to \mathbf{R}$. For each $i = 1, \ldots, K$, we introduce variables $h_i \in \mathbf{R}$ (to act as function values $h(x_i)$) and $g_i \in \mathbf{R}^n$ (to act as subgradients in $\partial h(x_i)$). We define

$$h(x) = \max_i \{h_i + g_i^T(x - z_i)\}.$$

This function is clearly convex, and we see that the first-order convexity conditions become $h_j \geq h_i + g_i^T(z_j - z_i)$ for all pairs (i, j), which are convex constraints in h, g. As we wish our function to be an (approximate) lower bound on f, we introduce the constraint $y_i \geq h_i$ for all i as well. Then, to obtain our approximate function h, we solve the convex optimization problem

$$\begin{array}{ll} \underset{h,g}{\text{minimize}} & \displaystyle\sum_{i=1}^{K} (h_i - y_i)^2 \\ \text{subject to} & h_j \geq h_i + g_j^T(z_j - z_i), \quad h_i \leq y_i \text{ for } i, j = 1, \ldots, K. \end{array} \tag{7}$$

In Figure 2, we give an example of this method's performance. In the plots, we show the results of the function $h(x)$ fit by the optimization problem (7), where the true function (the quartic $f(x) = x^4 - 2x^3 + .3x$) is shown as the solid blue line, with the sampled points and fitted function as the dotted black line. The left two plots ((a) and (b)) show the results of randomly chosen points, while (c) shows the results of choosing points equi-spaced over the interval $\mathcal{T}^{(k)} = [-2, 1]$.

An alternative, if the functions are smoother, is to fit quadratic functions to the data, which may have better performance. In this case, we will use $h(x) = \frac{1}{2}(x - x^{(k)})^T P(x - x^{(k)}) + q^T(x - x^{(k)}) + r$ as the function being fit, and constraining $P \succeq 0$, the closest (in ℓ_2 error) convex quadratic to the observed data is found by solving

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{K} \left((z_i - x^{(k)})^T P(z_i - x^{(k)}) + q^T(z_i - x^{(k)}) + r - y_i\right)^2 \\ \text{subject to} & P \succeq 0 \end{array}$$

over the variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$. This will not yield (even in the limit of infinite samples) a lower bound or upper bound on the function f, even locally, but it can be much more efficient from a sample requirement perspective than the piecewise affine fitting (7).

3.2 Composite optimization

A general family of nonconvex and nonsmooth functions are the so-called *convex composite* functions, which are functions of the form

$$f(x) = h(c(x)), \tag{8}$$

where $h : \mathbf{R}^m \to \mathbf{R}$ is convex and $c : \mathbf{R}^n \to \mathbf{R}^m$ is a smooth (differentiable) function. There are numerous applications of such functions.

The first applications arose out of the *exact penalty* approach, which we mentioned already in Section 2.5. In particular, if we consider the potentially nonconvex problem

$$\text{minimize} \;\; f(x) \quad \text{subject to} \quad c(x) = 0, \tag{9}$$

where the constraint $c(x) = 0$ is given by a continuously differentiable function $c : \mathbf{R}^n \to \mathbf{R}^m$, $m \le n$, then the exact penalty formulation of problem (9) is

$$\text{minimize} \;\; f(x) + \lambda \left\| c(x) \right\|,$$

where $\lambda \ge 0$ is a penalty parameter and $\|\cdot\|$ is some norm on $\mathbf{R}^m$. Under various constraint qualifications, this penalization leaves intact the set of local minimizers for problem (9). Indeed, suppose that x_0 is a local minimizer of problem (9), so that $f(x) \ge f(x_0)$ for all x near x_0 with $c(x) = 0$, and that (i) f is Lipschitz in a neighborhood of x_0 and (ii) the Jacobian transpose $\nabla c(x) = [\nabla c_1(x) \;\cdots\; \nabla c_m(x)] \in \mathbf{R}^{n \times m}$ has independent columns at x_0. Then in a neighborhood of x_0, $\nabla c(x)$ has independent columns, and we have for large enough λ that

$$\begin{aligned} f(x) + \lambda \left\| c(x) \right\| &\ge f(x_0) - \mathrm{Lip}(f) \left\| x - x_0 \right\| + \lambda \left\| c(x_0) + \nabla c(x_0)^T (x - x_0) + o(\|x - x_0\|) \right\| \\ &\ge f(x_0) + (\lambda \gamma_{\min}(\nabla c(x_0)) - \mathrm{Lip}(f)) \left\| x - x_0 \right\| + o(\|x - x_0\|) \\ &\ge f(x_0) = f(x_0) + \lambda \left\| c(x_0) \right\| \end{aligned}$$

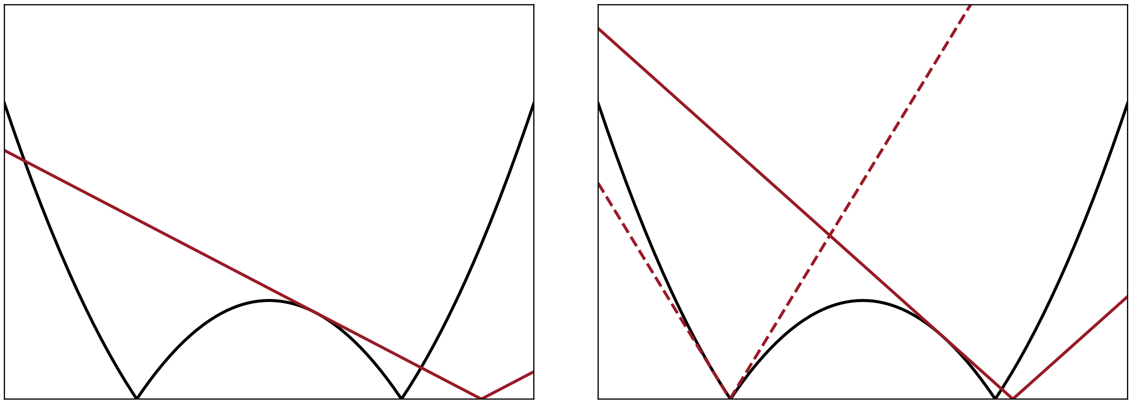


Figure 3. The function $f(x) = |x^2 - 1|$, along with convex approximations to it, local to the point x, given by $f_x(y) = |x^2 + 2x(y - x) - 1|$. This is the composite model of f given by Eq. (10).

for all x near enough x_0, where $\gamma_{\min}$ denotes the minimum singular value of its argument.

The formulation (8) has many other applications. For example, consider a machine learning or statistical application where we have data in pairs (x_i, y_i), and we wish to model y_i by some nonlinear but smooth function $\sigma(w^T x_i)$, where $W \in \mathbf{R}^{m \times n}$ is a matrix and $\sigma : \mathbf{R} \to \mathbf{R}$. If we place a convex loss ℓ on the error $y_i - \sigma(w^T x_i)$, we have optimization problem

$$f(w) = \sum_{i=1}^{N} \ell(y_i - \sigma(w^T x_i)),$$

which is the composition of a convex function ℓ with a smooth function σ. We will give more examples in the coming sections.

The methods for problem (8) are elegant and often very effective. As is typical in our optimization problems, our first step is to model the function f locally by some simpler to solve or minimize convex model. In this case, because c is smooth, a natural model for f in a neighborhood of x, which we denote by f_x to indicate its locality to x, is

$$f_x(y) := h(c(x) + \nabla c(x)^T (y - x)), \tag{10}$$

which is convex in y. See Figure 3. This then gives rise to the *prox-linear* method, which iterates

$$x^{(k+1)} = \operatorname*{argmin}_x \left\{ f_{x^{(k)}}(x) + \frac{1}{2\alpha_k} \left\| x - x^{(k)} \right\|^2 \right\}, \tag{11}$$

where $\alpha_k > 0$ is a stepsize, minimizing a regularized model of the function f centered around the point $x^{(k)}$.

There are a number of ways to choose the stepsize α_k. One of the simplest is the following: if the function h is L-Lipschitz and ∇c is β-Lipschitz, then a Taylor approximation shows that

$$\left\| c(y) - \left(c(x) + \nabla c(x)^T (y - x)\right) \right\| \le \frac{\beta}{2} \left\| y - x \right\|^2,$$

and thus

$$f_x(y) = h(c(x) + \nabla c(x)^T (y - x)) \le h(c(y)) + \frac{L\beta}{2} \left\| y - x \right\|^2,$$

and so any $\alpha_k \le \frac{1}{L\beta}$ guarantees decrease in the objective. A second common approach is to adapt the backtracking line search strategy of Section 2.5. In this case, if x_α^+ minimizes $f_x(y) + \frac{1}{2\alpha} \left\| y - x \right\|^2$ over y, we backtrack until the predicted decrease in objective is similar to the actual decrease in objective. Beginning from $t = 1$, until

$$f(x_{t\alpha}^+) \le f(x) - \kappa t \left[f(x) - f_x(x_{t\alpha}^+) \right]$$

for some $\kappa \in (0, 1)$, we divide $t := t/\beta$ for some $\beta > 1$. That is, we repeatedly minimize problem (11) at decreasing stepsizes until we make progress sufficiently close to the predicted progress by the approximation f_x. Alternatively, we may employ the simpler backtracking search at the conclusion of Section 2.5.

4 Examples

4.1 Nonlinear optimal control

TO WRITE

4.2 Phase retrieval

TO WRITE

5 Convergence of majorization-minimization

In this section, we give a brief argument typical of the type of convergence guarantees one can show using majorization-minimization schemes. We give a simple form of the argument, noting that there are substantially more possibilities and advanced convergence results [Bur85, BF95, DL18]. In the generality we consider, the best convergence one can hope for is convergence to a stationary point, which is what we show. To set notation, we look at the problem

$$\text{minimize } f(x)$$

where f is potentially non-convex, and it may include terms such as $\mathbf{I}_C(x)$ where C is convex, the $+\infty$-valued indicator of $x \in C$.

Before continuing, we give a few preliminaries, which include the class of functions we consider. We say that a function $f : \mathbf{R}^n \to \mathbf{R}$ is λ-*semi-convex* if for all $x_0 \in \mathbf{R}^n$, the function

$$f(x) + \frac{\lambda}{2} \left\| x - x_0 \right\|_2^2 \tag{12}$$

is convex. You should convince yourself that the choice of x_0 does not matter: if f is semi-convex for one x_0, it is semi-convex for all $x_0 \in \mathbf{R}^n$. (There is no standard name for this class of functions; it is variously called weak convexity, lower $\mathcal{C}^2$, and sometimes has names such as prox-regularity attached to it. See, e.g. [BDLM09, RW98, DL18].) With this class of functions, we can define an extended notion of subdifferential (called the Fréchet subdifferential), and while there are vastly more sophisticated ways to derive such extended subdifferentials, we follow the simplest attack and simply define

$$\partial f(x) := \partial_y \left\{ f(y) + \frac{\lambda}{2} \|y - x\|_2^2 \right\} \Big|_{y=x},$$

that is, $\partial f(x)$ is the subdifferential of the convex function (12) when we take $x_0 = x$ evaluated at x. A calculation shows that with this definition, we also have

$$\partial f(x) = \left\{ g \in \mathbf{R}^n \mid f(y) \ge f(x) + g^T(y - x) + O(\|y - x\|^2) \text{ as } y \to x \right\}.$$

One class of functions that satisfies this are the convex composite functions of Section 3.2. Indeed, let $f(x) = h(c(x))$, where $h : \mathbf{R}^m \to \mathbf{R}$ is convex and L-Lipschitz continuous and $c : \mathbf{R}^n \to \mathbf{R}^m$ has β-Lipschitz continuous gradient, that is,

$$|||\nabla c(x) - \nabla c(y)|||_{\mathrm{op}} \le \beta \|x - y\|_2,$$

where $|||\cdot|||_{\mathrm{op}}$ denotes the ℓ_2-operator norm (maximum singular value). Then we claim the following lemma.

Lemma 5.1. *Let $f = h \circ c$ as above. Then f is $L\beta$-semi-convex.*

Proof As h is a closed convex function, it is equal to its own biconjugate, that is, for $w \in \mathbf{R}^m$ we have

$$h(w) = \sup_v \left\{ v^T w - h^*(v) \right\} = \sup_{\|v\|_2 \le L} \left\{ v^T w - h^*(v) \right\},$$

where we have used that h is L-Lipschitz. Now, note that

$$x \mapsto w^T c(x) \;\text{ has }\; \nabla_x w^T c(x) = \nabla c(x) w,$$

which satisfies $\|\nabla c(x) w - \nabla c(y) w\|_2 \le L\beta \|x - y\|_2$, so that the second-order conditions for convexity imply

$$x \mapsto w^T c(x) + \frac{L\beta}{2} \|x - x_0\|_2^2$$

is convex for any $x_0 \in \mathbf{R}^n$. We then have

$$h(c(x)) + \frac{L\beta}{2} \|x - x_0\|_2^2 = \sup_{\|v\|_2 \le L} \left\{ v^T c(x) + \frac{L\beta}{2} \|x - x_0\|_2^2 - h^*(v) \right\},$$

which is the supremum of convex functions in x and hence convex. □

In any case, let us now consider a majorization minimization scheme, which at iteration k minimizes a *convex* function, centered at $x^{(k)}$, denoted $f_{x^{(k)}}$, with regularization. We assume that f is λ-semi-convex, and that there exists $\gamma \in \mathbf{R}_+$ such that for all $x \in \mathbf{R}^n$, we have the approximation guarantee

$$|f_x(y) - f(y)| \le \frac{\gamma}{2} \|y - x\|_2^2 . \tag{13}$$

(This may be satisfied for some $\gamma' < \gamma$, but we only consider γ.) The generic majorization-minimization scheme then iterates

$$x^{(k+1)} = \operatorname*{argmin}_x \left\{ f_{x^{(k)}}(x) + \frac{\gamma}{2} \left\| x - x^{(k)} \right\|_2^2 \right\} . \tag{14}$$

5.1 Nearly stationary points

The iteration (14), coupled with the semi-convexity condition (12), yields several interesting behaviors. To understand convergence to stationary points, we actually begin with the semi-convexity condition (12). Let us define the proximal point

$$x^{\mathrm{pp}} := \mathbf{prox}_{f/(2\lambda)}(x) = \operatorname*{argmin}_y \left\{ f(y) + \frac{\lambda}{2} \|y - x\|_2^2 + \frac{\lambda}{2} \|y - x\|_2^2 \right\} .$$

Without any loss of generality, we assume that $\lambda \ge \gamma$, as increasing λ simply makes the problem more strongly convex. Recall that $f(y) + (\lambda/2) \|y - x\|_2^2$ is convex, so the objective above is λ-strongly convex.[1] Using the standard optimality conditions for minimization of a convex function, we have for some

$$g \in \partial\{f(x^{\mathrm{pp}}) + \frac{\lambda}{2} \|x^{\mathrm{pp}} - x\|_2^2\} = \partial f(x^{\mathrm{pp}}) + \lambda(x^{\mathrm{pp}} - x),$$

where the second ∂ is the Fréchet subdifferential, that

$$g + \lambda(x^{\mathrm{pp}} - x) = 0. \tag{15}$$

Thus we have the optimality (or growth) guarantee

$$\begin{aligned} f(y) + \frac{\lambda}{2} \|y - x\|_2^2 &\ge f(x^{\mathrm{pp}}) + \frac{\lambda}{2} \|x^{\mathrm{pp}} - x\|_2^2 + g^T (y - x^{\mathrm{pp}}) \\ &= f(x^{\mathrm{pp}}) + \frac{\lambda}{2} \|x^{\mathrm{pp}} - x\|_2^2 - \lambda (x - x^{\mathrm{pp}})^T (y - x^{\mathrm{pp}}) \\ &= f(x^{\mathrm{pp}}) + \lambda \|x - x^{\mathrm{pp}}\|_2^2 + \frac{\lambda}{2} \|y - x^{\mathrm{pp}}\|_2^2 - \frac{\lambda}{2} \|y - x\|_2^2 , \end{aligned}$$

or

$$f(y) + \lambda \|y - x\|_2^2 \ge f(x^{\mathrm{pp}}) + \lambda \|x^{\mathrm{pp}} - x\|_2^2 + \frac{\lambda}{2} \|y - x^{\mathrm{pp}}\|_2^2 \tag{16}$$

for all $y \in \mathbf{R}^n$. Summarizing, by using equality (15), we see that

[1] A convex function h is λ-strongly convex if $h(y) \ge h(x) + g^T(y - x) + \frac{\lambda}{2} \|y - x\|_2^2$ for all x, y and $g \in \partial f(x)$.

Lemma 5.2. *Suppose that x satisfies the proximal-point closeness condition $\|x^{\text{pp}} - x\|_2 \le \epsilon$. Then*

- *(i)* Near stationarity*: The proximal point x^{pp} guarantees there exists $g \in \partial f(x^{\text{pp}})$ such that $\|g\|_2 \le \lambda\epsilon$, or $\operatorname{dist}(0, \partial f(x^{\text{pp}})) \le \lambda\epsilon$*
- *(ii)* Improvement*: The proximal point x^{pp} satisfies $f(x^{\text{pp}}) \le f(x)$.*

That is, if the proximal point update is small, then it is nearly stationary. This suggests that if our model-based update (14) is small, then the resulting point should somehow be nearly stationary as well. This is indeed our strategy.

Let us return to the iteration (14). We show that if the general update

$$x^+ = \operatorname*{argmin}_y \left\{ f_x(y) + \frac{\gamma}{2} \|y - x\|_2^2 \right\}.$$

leaves x^+ near x, then the points x^+ and x^{pp} from the proximal point update are also close, yielding a type of near-stationarity for x^+. Following an identical derivation to obtain inequality (14), we have

$$f_x(x^+) + \frac{\gamma}{2} \left\|x^+ - x\right\|_2^2 + \frac{\gamma}{2} \left\|y - x^+\right\|_2^2 \le f_x(y) + \frac{\gamma}{2} \|y - x\|_2^2 \tag{17}$$

Now, substitute $y = x^{\text{pp}}$ in the preceding inequality to obtain

$$f_x(x^+) + \frac{\gamma}{2} \left\|x^+ - x\right\|_2^2 + \frac{\gamma}{2} \left\|x^{\text{pp}} - x^+\right\|_2^2 \le f_x(x^{\text{pp}}) + \frac{\gamma}{2} \|x^{\text{pp}} - x\|_2^2 .$$

Using the approximation condition (13), we immediately obtain that

$$\begin{aligned}
f(x^+) + \frac{\gamma}{2} \left\|x^{\text{pp}} - x^+\right\|_2^2 &\le f_x(x^+) + \frac{\gamma}{2} \left\|x^+ - x\right\|_2^2 + \frac{\gamma}{2} \left\|x^{\text{pp}} - x^+\right\|_2^2 \\
&\le f(x^{\text{pp}}) + \gamma \|x^{\text{pp}} - x\|_2^2 \\
&= f(x^{\text{pp}}) + \lambda \|x^{\text{pp}} - x\|_2^2 + (\gamma - \lambda) \|x^{\text{pp}} - x\|_2^2 \\
&\le f(x^+) + \lambda \left\|x^+ - x\right\|_2^2 + (\gamma - \lambda) \|x^{\text{pp}} - x\|_2^2 - \frac{\lambda}{2} \left\|x^+ - x^{\text{pp}}\right\|_2^2 ,
\end{aligned}$$

where in the final inequality we used that x^{pp} minimizes the λ-strongly convex function $f(y) + \lambda \|y - x\|_2^2$ over y. Rearranging, and using our assumption that $\lambda \ge \gamma$, we have

$$\frac{\gamma + \lambda}{2} \left\|x^+ - x^{\text{pp}}\right\|_2^2 \le \lambda \left\|x^+ - x\right\|_2^2 .$$

Summarizing, we have the following three results, which show that if $\|x^+ - x\|$ is small, then there exists a point $\widehat{x}$ near x^+ that is well-behaved. (Take $\widehat{x} = x^{\text{pp}}$ and use Lemma 5.2.)

Lemma 5.3. *Let $x^+ = \operatorname{argmin}_y \{f_x(y) + \frac{\gamma}{2} \|y - x\|_2^2\}$. Then there exists $\widehat{x}$ satisfying the following three conditions:*

(i) Point proximity*: we have*

$$\|\widehat{x} - x^+\|_2 \le \sqrt{\frac{2\lambda}{\gamma + \lambda}} \left\|x^+ - x\right\|_2$$

(ii) Value proximity*: we have*

$$f(\widehat{x}) \le f(x^+) + \lambda \left\|x^+ - x\right\|_2^2$$

(iii) Near stationarity*:*

$$\operatorname{dist}(0, \partial f(\widehat{x})) \le \lambda \sqrt{\frac{2\lambda}{\gamma + \lambda}} \left\|x^+ - x\right\|_2.$$

We can restate the conditions above in a slightly different way: whenever the iterates $x^{(k)}$ and $x^{(k+1)}$ are suitably close, then $x^{(k+1)}$ is near a point which is nearly stationary. This justifies a common heuristic: we simply stop when $\left\|x^{(k+1)} - x^{(k)}\right\|$ is small. In general, we use a better normalized version of this quantity, known as the *gradient mapping*, which is defined by

$$\mathsf{G}_\gamma(x) := \gamma(x - x^+), \tag{18}$$

which we see guarantees that the iterates satisfy $x^{(k+1)} = x^{(k)} - \alpha \mathsf{G}_\gamma(x^{(k)})$ for the stepsize $\alpha = 1/\gamma$, justifying the term gradient mapping. We then stop the iterations when

$$\left\|\mathsf{G}_{1/\alpha}(x^{(k)})\right\|_2 \le \epsilon.$$

5.2 Convergence of majorization-minimization

With our guarantees that small gradient mapping (18) guarantees nearly stationary points, we finally return to prove that the majorization-minimization scheme actually yields points that have small gradient mapping. We assume that there exists $x^\star$ satisfying $f(x^\star) = \inf_x f(x)$, though this is stronger than necessary but simplifies notation. Consider the update scheme (14). First, we see that the iterates yield non-increasing function values, as we always have

$$f(x^{(k+1)}) \le f_{x^{(k)}}(x^{(k+1)}) + \frac{\gamma}{2} \left\|x^{(k+1)} - x^{(k)}\right\|_2^2 \le f_{x^{(k)}}(x^{(k)}) = f(x^{(k)})$$

by the approximation condition (13). Second, because $f_x(y)$ is convex in y by assumption, we can make a stronger progress guarantee. Indeed, the objective (14) is γ-strongly convex in x, so as in the previous section's Eq. (17), we have the stronger progress guarantee that

$$\begin{aligned} f(x^{(k+1)}) &\le f_{x^{(k)}}(x^{(k+1)}) + \frac{\gamma}{2} \left\|x^{(k+1)} - x^{(k)}\right\|_2^2 \\ &\le f_{x^{(k)}}(x^{(k)}) - \frac{\gamma}{2} \left\|x^{(k+1)} - x^{(k)}\right\|_2^2 = f(x^{(k)}) - \frac{\gamma}{2} \left\|x^{(k+1)} - x^{(k)}\right\|_2^2. \end{aligned}$$

Rearranging and summing, we have

$$\sum_{i=1}^{k} \frac{\gamma}{2} \left\|x^{(i+1)} - x^{(i)}\right\|_2^2 \le \sum_{i=1}^{k} \left[f(x^{(i)}) - f(x^{(i+1)})\right] = f(x^{(1)}) - f(x^{(k+1)}) \le f(x^{(1)}) - f(x^\star),$$

where the final inequality uses that $f(x^{(k)})$ is non-increasing and that $f(x^\star) \le f(x^{(k)})$ for all k. Using the definition (18) of the gradient mapping, we see that

$$\sum_{i=1}^{k} \left\|\mathsf{G}_\gamma(x^{(i)})\right\|_2^2 \le 2\gamma \left[f(x^{(1)}) - f(x^\star)\right],$$

and

$$\min_{i \le k} \left\|\mathsf{G}_\gamma(x^{(i)})\right\|_2^2 \le \frac{2\gamma[f(x^{(1)}) - f(x^\star)]}{k}.$$

Additional reading and notes

There are numerous references on this material, most of which is classical. The books of Conn, Gould, and Toint [CGT00] and Nocedal and Wright [NW06] contain numerous results on trust region methods and local methods, such as Newton's method applied on non-convex problems. Bertsekas's book [Ber99] contains a wealth of material on general nonlinear optimization problems. The cubic-regularized Newton method was discovered by Griewank [Gri81] in an unpublished technical report and rediscovered, and used for finding stationary points of smooth nonconvex functions, in [NP06]. The composite optimization approach has a long history in optimization, dating back at least to Fletcher's work on exact-penalty formulations for generic nonlinear programming problems [Fle82, FW80, Fle87]. There has been substantial development in the intervening years; a few references include [Bur85, BF95, DL18].

References

[BDLM09] Jérôme Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2009.

[Ber99] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

[BF95] James Burke and Michael Ferris. A Gauss-Newton method for convex composite optimization. *Mathematical Programming*, 71:179–194, 1995.

[Bur85] James Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33:260–279, 1985.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[CGT00] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods.* MPS-SIAM Series on Optimization. SIAM, 2000.

[DL18] Dmitriy Drusvyatskiy and Adrian Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, To appear, 2018.

[Fle82] Roger Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Study*, 17:67–76, 1982.

[Fle87] R. Fletcher. *Practical Methods of Optimization.* John Wiley, second edition, 1987.

[FW80] Roger Fletcher and G. Alistair Watson. First and second order conditions for a class of nondifferentiable optimization problems. *Mathematical Programming*, 18:291–307, 1980.

[Gri81] Andreas Griewank. The modification of Newtons method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[NP06] Yurii Nesterov and Boris Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108:177–205, 2006.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, 2006.

[RW98] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis.* Springer, New York, 1998.