Chapter 9

Unconstrained minimization

9.1 Unconstrained minimization problems

In this chapter we discuss methods for solving the unconstrained optimization problem

$$\text{minimize} \quad f(x) \tag{9.1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is convex and twice continuously differentiable (which implies that $\mathbf{dom}\, f$ is open). We will assume that the problem is solvable, *i.e.*, there exists an optimal point $x^\star$. (More precisely, the assumptions later in the chapter will imply that $x^\star$ exists and is unique.) We denote the optimal value, $\inf_x f(x) = f(x^\star)$, as $p^\star$.

Since f is differentiable and convex, a necessary and sufficient condition for a point $x^\star$ to be optimal is

$$\nabla f(x^\star) = 0 \tag{9.2}$$

(see §4.2.3). Thus, solving the unconstrained minimization problem (9.1) is the same as finding a solution of (9.2), which is a set of n equations in the n variables $x_1, \ldots, x_n$. In a few special cases, we can find a solution to the problem (9.1) by analytically solving the optimality equation (9.2), but usually the problem must be solved by an iterative algorithm. By this we mean an algorithm that computes a sequence of points $x^{(0)}, x^{(1)}, \ldots \in \mathbf{dom}\, f$ with $f(x^{(k)}) \to p^\star$ as $k \to \infty$. Such a sequence of points is called a *minimizing sequence* for the problem (9.1). The algorithm is terminated when $f(x^{(k)}) - p^\star \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance.

Initial point and sublevel set

The methods described in this chapter require a suitable starting point $x^{(0)}$. The starting point must lie in $\mathbf{dom}\, f$, and in addition the sublevel set

$$S = \{x \in \mathbf{dom}\, f \mid f(x) \leq f(x^{(0)})\} \tag{9.3}$$

must be closed. This condition is satisfied for all $x^{(0)} \in \mathbf{dom}\, f$ if the function f is *closed*, *i.e.*, all its sublevel sets are closed (see §A.3.3). Continuous functions with

$\mathbf{dom}\, f = \mathbf{R}^n$ are closed, so if $\mathbf{dom}\, f = \mathbf{R}^n$, the initial sublevel set condition is satisfied by any $x^{(0)}$. Another important class of closed functions are continuous functions with open domains, for which $f(x)$ tends to infinity as x approaches $\mathbf{bd}\,\mathbf{dom}\, f$.

9.1.1 Examples

Quadratic minimization and least-squares

The general convex quadratic minimization problem has the form

$$\text{minimize} \quad (1/2)x^T Px + q^T x + r, \tag{9.4}$$

where $P \in \mathbf{S}^n_+$, $q \in \mathbf{R}^n$, and $r \in \mathbf{R}$. This problem can be solved via the optimality conditions, $Px^\star + q = 0$, which is a set of linear equations. When $P \succ 0$, there is a unique solution, $x^\star = -P^{-1}q$. In the more general case when P is not positive definite, any solution of $Px^\star = -q$ is optimal for (9.4); if $Px^\star = -q$ does not have a solution, then the problem (9.4) is unbounded below (see exercise 9.1). Our ability to analytically solve the quadratic minimization problem (9.4) is the basis for Newton's method, a powerful method for unconstrained minimization described in §9.5.

One special case of the quadratic minimization problem that arises very frequently is the least-squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2 = x^T(A^TA)x - 2(A^Tb)^Tx + b^Tb.$$

The optimality conditions

$$A^TAx^\star = A^Tb$$

are called the *normal equations* of the least-squares problem.

Unconstrained geometric programming

As a second example, we consider an unconstrained geometric program in convex form,

$$\text{minimize} \quad f(x) = \log\left(\sum_{i=1}^m \exp(a_i^Tx + b_i)\right).$$

The optimality condition is

$$\nabla f(x^\star) = \frac{1}{\sum_{j=1}^m \exp(a_j^Tx^\star + b_j)} \sum_{i=1}^m \exp(a_i^Tx^\star + b_i)a_i = 0,$$

which in general has no analytical solution, so here we must resort to an iterative algorithm. For this problem, $\mathbf{dom}\, f = \mathbf{R}^n$, so any point can be chosen as the initial point $x^{(0)}$.

Analytic center of linear inequalities

We consider the optimization problem

$$\text{minimize} \quad f(x) = -\sum_{i=1}^m \log(b_i - a_i^Tx), \tag{9.5}$$

where the domain of f is the open set

$$\mathbf{dom}\, f = \{x \mid a_i^T x < b_i,\ i = 1, \ldots, m\}.$$

The objective function f in this problem is called the *logarithmic barrier* for the inequalities $a_i^T x \leq b_i$. The solution of (9.5), if it exists, is called the *analytic center* of the inequalities. The initial point $x^{(0)}$ must satisfy the strict inequalities $a_i^T x^{(0)} < b_i$, $i = 1, \ldots, m$. Since f is closed, the sublevel set S for any such point is closed.

Analytic center of a linear matrix inequality

A closely related problem is

$$\text{minimize} \quad f(x) = \log \det F(x)^{-1} \tag{9.6}$$

where $F : \mathbf{R}^n \to \mathbf{S}^p$ is affine, *i.e.*,

$$F(x) = F_0 + x_1 F_1 + \cdots + x_n F_n,$$

with $F_i \in \mathbf{S}^p$. Here the domain of f is

$$\mathbf{dom}\, f = \{x \mid F(x) \succ 0\}.$$

The objective function f is called the *logarithmic barrier* for the linear matrix inequality $F(x) \succeq 0$, and the solution (if it exists) is called the analytic center of the linear matrix inequality. The initial point $x^{(0)}$ must satisfy the strict linear matrix inequality $F(x^{(0)}) \succ 0$. As in the previous example, the sublevel set of any such point will be closed, since f is closed.

9.1.2 Strong convexity and implications

In much of this chapter (with the exception of §9.6) we assume that the objective function is *strongly convex* on S, which means that there exists an $m > 0$ such that

$$\nabla^2 f(x) \succeq mI \tag{9.7}$$

for all $x \in S$. Strong convexity has several interesting consequences. For $x, y \in S$ we have

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

for some z on the line segment $[x, y]$. By the strong convexity assumption (9.7), the last term on the righthand side is at least $(m/2)\|y - x\|_2^2$, so we have the inequality

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2 \tag{9.8}$$

for all x and y in S. When $m = 0$, we recover the basic inequality characterizing convexity; for $m > 0$ we obtain a better lower bound on $f(y)$ than follows from convexity alone.

We will first show that the inequality (9.8) can be used to bound $f(x) - p^\star$, which is the suboptimality of the point x, in terms of $\|\nabla f(x)\|_2$. The righthand side of (9.8) is a convex quadratic function of y (for fixed x). Setting the gradient with respect to y equal to zero, we find that $\tilde{y} = x - (1/m)\nabla f(x)$ minimizes the righthand side. Therefore we have

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|_2^2 \\ &\geq f(x) + \nabla f(x)^T(\tilde{y}-x) + \frac{m}{2}\|\tilde{y}-x\|_2^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2. \end{aligned}$$

Since this holds for any $y \in S$, we have

$$p^\star \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2. \tag{9.9}$$

This inequality shows that if the gradient is small at a point, then the point is nearly optimal. The inequality (9.9) can also be interpreted as a condition for *suboptimality* which generalizes the optimality condition (9.2):

$$\|\nabla f(x)\|_2 \leq (2m\epsilon)^{1/2} \Longrightarrow f(x) - p^\star \leq \epsilon. \tag{9.10}$$

We can also derive a bound on $\|x - x^\star\|_2$, the distance between x and any optimal point $x^\star$, in terms of $\|\nabla f(x)\|_2$:

$$\|x - x^\star\|_2 \leq \frac{2}{m}\|\nabla f(x)\|_2. \tag{9.11}$$

To see this, we apply (9.8) with $y = x^\star$ to obtain

$$\begin{aligned} p^\star = f(x^\star) &\geq f(x) + \nabla f(x)^T(x^\star - x) + \frac{m}{2}\|x^\star - x\|_2^2 \\ &\geq f(x) - \|\nabla f(x)\|_2\|x^\star - x\|_2 + \frac{m}{2}\|x^\star - x\|_2^2, \end{aligned}$$

where we use the Cauchy-Schwarz inequality in the second inequality. Since $p^\star \leq f(x)$, we must have

$$-\|\nabla f(x)\|_2\,\|x^\star - x\|_2 + \frac{m}{2}\|x^\star - x\|_2^2 \leq 0,$$

from which (9.11) follows. One consequence of (9.11) is that the optimal point $x^\star$ is unique.

Upper bound on $\nabla^2 f(x)$

The inequality (9.8) implies that the sublevel sets contained in S are bounded, so in particular, S is bounded. Therefore the maximum eigenvalue of $\nabla^2 f(x)$, which is a continuous function of x on S, is bounded above on S, *i.e.*, there exists a constant M such that

$$\nabla^2 f(x) \preceq MI \tag{9.12}$$

for all $x \in S$. This upper bound on the Hessian implies for any x, $y \in S$,

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{M}{2}\|y-x\|_2^2, \tag{9.13}$$

which is analogous to (9.8). Minimizing each side over y yields

$$p^\star \leq f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2, \tag{9.14}$$

the counterpart of (9.9).

Condition number of sublevel sets

From the strong convexity inequality (9.7) and the inequality (9.12), we have

$$mI \preceq \nabla^2 f(x) \preceq MI \tag{9.15}$$

for all $x \in S$. The ratio $\kappa = M/m$ is thus an upper bound on the condition number of the matrix $\nabla^2 f(x)$, *i.e.*, the ratio of its largest eigenvalue to its smallest eigenvalue. We can also give a geometric interpretation of (9.15) in terms of the sublevel sets of f.

We define the *width* of a convex set $C \subseteq \mathbf{R}^n$, in the direction q, where $\|q\|_2 = 1$, as

$$W(C,q) = \sup_{z \in C} q^T z - \inf_{z \in C} q^T z.$$

The *minimum width* and *maximum width* of C are given by

$$W_{\min} = \inf_{\|q\|_2=1} W(C,q), \qquad W_{\max} = \sup_{\|q\|_2=1} W(C,q).$$

The *condition number* of the convex set C is defined as

$$\mathbf{cond}(C) = \frac{W_{\max}^2}{W_{\min}^2},$$

i.e., the square of the ratio of its maximum width to its minimum width. The condition number of C gives a measure of its *anisotropy* or *eccentricity*. If the condition number of a set C is small (say, near one) it means that the set has approximately the same width in all directions, *i.e.*, it is nearly spherical. If the condition number is large, it means that the set is far wider in some directions than in others.

Example 9.1 *Condition number of an ellipsoid.* Let $\mathcal{E}$ be the ellipsoid

$$\mathcal{E} = \{x \mid (x-x_0)^T A^{-1}(x-x_0) \leq 1\},$$

where $A \in \mathbf{S}^n_{++}$. The width of $\mathcal{E}$ in the direction q is

$$\begin{aligned} \sup_{z \in \mathcal{E}} q^T z - \inf_{z \in \mathcal{E}} q^T z &= (\|A^{1/2}q\|_2 + q^T x_0) - (-\|A^{1/2}q\|_2 + q^T x_0) \\ &= 2\|A^{1/2}q\|_2. \end{aligned}$$

It follows that its minimum and maximum width are

$$W_{\min} = 2\lambda_{\min}(A)^{1/2}, \qquad W_{\max} = 2\lambda_{\max}(A)^{1/2},$$

and its condition number is

$$\mathbf{cond}(\mathcal{E}) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \kappa(A),$$

where $\kappa(A)$ denotes the condition number of the matrix A, *i.e.*, the ratio of its maximum singular value to its minimum singular value. Thus the condition number of the ellipsoid $\mathcal{E}$ is the same as the condition number of the matrix A that defines it.

Now suppose f satisfies $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in S$. We will derive a bound on the condition number of the α-sublevel $C_\alpha = \{x \mid f(x) \leq \alpha\}$, where $p^\star < \alpha \leq f(x^{(0)})$. Applying (9.13) and (9.8) with $x = x^\star$, we have

$$p^\star + (M/2)\|y - x^\star\|_2^2 \geq f(y) \geq p^\star + (m/2)\|y - x^\star\|_2^2.$$

This implies that $B_{\text{inner}} \subseteq C_\alpha \subseteq B_{\text{outer}}$ where

$$\begin{aligned} B_{\text{inner}} &= \{y \mid \|y - x^\star\|_2 \leq (2(\alpha - p^\star)/M)^{1/2}\}, \\ B_{\text{outer}} &= \{y \mid \|y - x^\star\|_2 \leq (2(\alpha - p^\star)/m)^{1/2}\}. \end{aligned}$$

In other words, the α-sublevel set contains B_{inner}, and is contained in B_{outer}, which are balls with radii

$$(2(\alpha - p^\star)/M)^{1/2}, \qquad (2(\alpha - p^\star)/m)^{1/2},$$

respectively. The ratio of the radii squared gives an upper bound on the condition number of C_α:

$$\mathbf{cond}(C_\alpha) \leq \frac{M}{m}.$$

We can also give a geometric interpretation of the condition number $\kappa(\nabla^2 f(x^\star))$ of the Hessian at the optimum. From the Taylor series expansion of f around $x^\star$,

$$f(y) \approx p^\star + \frac{1}{2}(y - x^\star)^T \nabla^2 f(x^\star)(y - x^\star),$$

we see that, for α close to $p^\star$,

$$C_\alpha \approx \{y \mid (y - x^\star)^T \nabla^2 f(x^\star)(y - x^\star) \leq 2(\alpha - p^\star)\},$$

i.e., the sublevel set is well approximated by an ellipsoid with center $x^\star$. Therefore

$$\lim_{\alpha \to p^\star} \mathbf{cond}(C_\alpha) = \kappa(\nabla^2 f(x^\star)).$$

We will see that the condition number of the sublevel sets of f (which is bounded by M/m) has a strong effect on the efficiency of some common methods for unconstrained minimization.

The strong convexity constants

It must be kept in mind that the constants m and M are known only in rare cases, so the inequality (9.10) cannot be used as a practical stopping criterion. It can be considered a *conceptual* stopping criterion; it shows that if the gradient of f at x is small enough, then the difference between $f(x)$ and $p^\star$ is small. If we terminate an algorithm when $\|\nabla f(x^{(k)})\|_2 \leq \eta$, where η is chosen small enough to be (very likely) smaller than $(m\epsilon)^{1/2}$, then we have $f(x^{(k)}) - p^\star \leq \epsilon$ (very likely).

In the following sections we give convergence proofs for algorithms, which include bounds on the number of iterations required before $f(x^{(k)}) - p^\star \leq \epsilon$, where ϵ is some positive tolerance. Many of these bounds involve the (usually unknown) constants m and M, so the same comments apply. These results are at least conceptually useful; they establish that the algorithm converges, even if the bound on the number of iterations required to reach a given accuracy depends on constants that are unknown.

We will encounter one important exception to this situation. In §9.6 we will study a special class of convex functions, called *self-concordant*, for which we can provide a complete convergence analysis (for Newton's method) that does not depend on any unknown constants.

9.2 Descent methods

The algorithms described in this chapter produce a minimizing sequence $x^{(k)}$, $k = 1, \ldots$, where

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$$

and $t^{(k)} > 0$ (except when $x^{(k)}$ is optimal). Here the concatenated symbols Δ and x that form Δx are to be read as a single entity, a vector in $\mathbf{R}^n$ called the *step* or *search direction* (even though it need not have unit norm), and $k = 0, 1, \ldots$ denotes the iteration number. The scalar $t^{(k)} \geq 0$ is called the *step size* or *step length* at iteration k (even though it is not equal to $\|x^{(k+1)} - x^{(k)}\|$ unless $\|\Delta x^{(k)}\| = 1$). The terms 'search step' and 'scale factor' are more accurate, but 'search direction' and 'step length' are the ones widely used. When we focus on one iteration of an algorithm, we sometimes drop the superscripts and use the lighter notation $x^+ = x + t\Delta x$, or $x := x + t\Delta x$, in place of $x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$.

All the methods we study are *descent methods*, which means that

$$f(x^{(k+1)}) < f(x^{(k)}),$$

except when $x^{(k)}$ is optimal. This implies that for all k we have $x^{(k)} \in S$, the initial sublevel set, and in particular we have $x^{(k)} \in \mathbf{dom}\, f$. From convexity we know that $\nabla f(x^{(k)})^T(y - x^{(k)}) \geq 0$ implies $f(y) \geq f(x^{(k)})$, so the search direction in a descent method must satisfy

$$\nabla f(x^{(k)})^T\Delta x^{(k)} < 0,$$

i.e., it must make an acute angle with the negative gradient. We call such a direction a *descent direction* (for f, at $x^{(k)}$).

The outline of a general descent method is as follows. It alternates between two steps: determining a descent direction Δx, and the selection of a step size t.

Algorithm 9.1 *General descent method.*

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

The second step is called the *line search* since selection of the step size t determines where along the line $\{x + t\Delta x \mid t \in \mathbf{R}_+\}$ the next iterate will be. (A more accurate term might be *ray search*.)

A practical descent method has the same general structure, but might be organized differently. For example, the stopping criterion is often checked while, or immediately after, the descent direction Δx is computed. The stopping criterion is often of the form $\|\nabla f(x)\|_2 \leq \eta$, where η is small and positive, as suggested by the suboptimality condition (9.9).

Exact line search

One line search method sometimes used in practice is *exact line search*, in which t is chosen to minimize f along the ray $\{x + t\Delta x \mid t \geq 0\}$:

$$t = \operatorname{argmin}_{s \geq 0} f(x + s\Delta x). \tag{9.16}$$

An exact line search is used when the cost of the minimization problem with one variable, required in (9.16), is low compared to the cost of computing the search direction itself. In some special cases the minimizer along the ray can be found analytically, and in others it can be computed efficiently. (This is discussed in §9.7.1.)

Backtracking line search

Most line searches used in practice are *inexact*: the step length is chosen to approximately minimize f along the ray $\{x + t\Delta x \mid t \geq 0\}$, or even to just reduce f 'enough'. Many inexact line search methods have been proposed. One inexact line search method that is very simple and quite effective is called *backtracking* line search. It depends on two constants α, β with $0 < \alpha < 0.5$, $0 < \beta < 1$.

Algorithm 9.2 *Backtracking line search.*

given a descent direction Δx for f at $x \in \mathbf{dom}\, f$, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$.
$t := 1$.
while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$, $t := \beta t$.

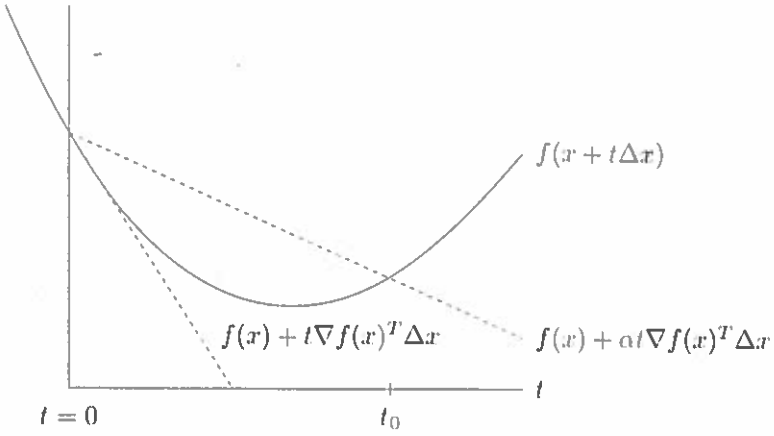

Figure 9.1 *Backtracking line search.* The curve shows f, restricted to the line over which we search. The lower dashed line shows the linear extrapolation of f, and the upper dashed line has a slope a factor of α smaller. The backtracking condition is that f lies below the upper dashed line, *i.e.*, $0 \leq t \leq t_0$.

The line search is called backtracking because it starts with unit step size and then reduces it by the factor β until the stopping condition $f(x+t\Delta x) \leq f(x) + \alpha t\nabla f(x)^T\Delta x$ holds. Since Δx is a descent direction, we have $\nabla f(x)^T\Delta x < 0$, so for small enough t we have

$$f(x+t\Delta x) \approx f(x) + t\nabla f(x)^T\Delta x < f(x) + \alpha t\nabla f(x)^T\Delta x,$$

which shows that the backtracking line search eventually terminates. The constant α can be interpreted as the fraction of the decrease in f predicted by linear extrapolation that we will accept. (The reason for requiring α to be smaller than 0.5 will become clear later.)

The backtracking condition is illustrated in figure 9.1. This figure suggests, and it can be shown, that the backtracking exit inequality $f(x+t\Delta x) \leq f(x) + \alpha t\nabla f(x)^T\Delta x$ holds for $t \geq 0$ in an interval $(0, t_0]$. It follows that the backtracking line search stops with a step length t that satisfies

$$t = 1, \qquad \text{or} \qquad t \in (\beta t_0, t_0].$$

The first case occurs when the step length $t = 1$ satisfies the backtracking condition, *i.e.*, $1 \leq t_0$. In particular, we can say that the step length obtained by backtracking line search satisfies

$$t \geq \min\{1, \beta t_0\}.$$

When $\mathbf{dom}\, f$ is not all of $\mathbf{R}^n$, the condition $f(x+t\Delta x) \leq f(x) + \alpha t\nabla f(x)^T\Delta x$ in the backtracking line search must be interpreted carefully. By our convention that f is infinite outside its domain, the inequality implies that $x + t\Delta x \in \mathbf{dom}\, f$. In a practical implementation, we first multiply t by β until $x + t\Delta x \in \mathbf{dom}\, f$:

then we start to check whether the inequality $f(x + t\Delta x) \leq f(x) + \alpha t \nabla f(x)^T \Delta x$ holds.

The parameter α is typically chosen between 0.01 and 0.3, meaning that we accept a decrease in f between 1% and 30% of the prediction based on the linear extrapolation. The parameter β is often chosen to be between 0.1 (which corresponds to a very crude search) and 0.8 (which corresponds to a less crude search).

9.3 Gradient descent method

A natural choice for the search direction is the negative gradient $\Delta x = -\nabla f(x)$. The resulting algorithm is called the *gradient algorithm* or *gradient descent method.*

Algorithm 9.3 *Gradient descent method.*

given a starting point $x \in \mathbf{dom}\, f$.

repeat

1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

The stopping criterion is usually of the form $\|\nabla f(x)\|_2 \leq \eta$, where η is small and positive. In most implementations, this condition is checked after step 1, rather than after the update.

9.3.1 Convergence analysis

In this section we present a simple convergence analysis for the gradient method, using the lighter notation $x^+ = x + t\Delta x$ for $x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$, where $\Delta x = -\nabla f(x)$. We assume f is strongly convex on S, so there are positive constants m and M such that $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in S$. Define the function $\tilde{f} : \mathbf{R} \to \mathbf{R}$ by $\tilde{f}(t) = f(x - t\nabla f(x))$, *i.e.*, f as a function of the step length t in the negative gradient direction. In the following discussion we will only consider t for which $x - t\nabla f(x) \in S$. From the inequality (9.13), with $y = x - t\nabla f(x)$, we obtain a quadratic upper bound on $\tilde{f}$:

$$\tilde{f}(t) \leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2. \tag{9.17}$$

Analysis for exact line search

We now assume that an exact line search is used, and minimize over t both sides of the inequality (9.17). On the lefthand side we get $\tilde{f}(t_{\text{exact}})$, where t_{exact} is the step length that minimizes $\tilde{f}$. The righthand side is a simple quadratic, which

is minimized by $t = 1/M$, and has minimum value $f(x) - (1/(2M))\|\nabla f(x)\|_2^2$. Therefore we have

$$f(x^+) = \tilde{f}(t_{\text{exact}}) \leq f(x) - \frac{1}{2M}\|\nabla(f(x))\|_2^2.$$

Subtracting $p^\star$ from both sides, we get

$$f(x^+) - p^\star \leq f(x) - p^\star - \frac{1}{2M}\|\nabla f(x)\|_2^2.$$

We combine this with $\|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^\star)$ (which follows from (9.9)) to conclude

$$f(x^+) - p^\star \leq (1 - m/M)(f(x) - p^\star).$$

Applying this inequality recursively, we find that

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star) \tag{9.18}$$

where $c = 1 - m/M < 1$, which shows that $f(x^{(k)})$ converges to $p^\star$ as $k \to \infty$. In particular, we must have $f(x^{(k)}) - p^\star \leq \epsilon$ after at most

$$\frac{\log((f(x^{(0)}) - p^\star)/\epsilon)}{\log(1/c)} \tag{9.19}$$

iterations of the gradient method with exact line search.

This bound on the number of iterations required, even though crude, can give some insight into the gradient method. The numerator,

$$\log((f(x^{(0)}) - p^\star)/\epsilon)$$

can be interpreted as the log of the ratio of the initial suboptimality (*i.e.*, gap between $f(x^{(0)})$ and $p^\star$), to the final suboptimality (*i.e.*, less than ϵ). This term suggests that the number of iterations depends on how good the initial point is, and what the final required accuracy is.

The denominator appearing in the bound (9.19), $\log(1/c)$, is a function of M/m, which we have seen is a bound on the condition number of $\nabla^2 f(x)$ over S, or the condition number of the sublevel sets $\{z \mid f(z) \leq \alpha\}$. For large condition number bound M/m, we have

$$\log(1/c) = -\log(1 - m/M) \approx m/M,$$

so our bound on the number of iterations required increases approximately linearly with increasing M/m.

We will see that the gradient method does in fact require a large number of iterations when the Hessian of f, near $x^\star$, has a large condition number. Conversely, when the sublevel sets of f are relatively isotropic, so that the condition number bound M/m can be chosen to be relatively small, the bound (9.18) shows that convergence is rapid, since c is small, or at least not too close to one.

The bound (9.18) shows that the error $f(x^{(k)}) - p^\star$ converges to zero at least as fast as a geometric series. In the context of iterative numerical methods, this is called *linear convergence*, since the error lies below a line on a log-linear plot of error versus iteration number.

Analysis for backtracking line search

Now we consider the case where a backtracking line search is used in the gradient descent method. We will show that the backtracking exit condition,

$$\tilde{f}(t) \leq f(x) - \alpha t\|\nabla f(x)\|_2^2,$$

is satisfied whenever $0 \leq t \leq 1/M$. First note that

$$0 \leq t \leq 1/M \implies -t + \frac{Mt^2}{2} \leq -t/2$$

(which follows from convexity of $-t+Mt^2/2$). Using this result and the bound (9.17), we have, for $0 \leq t \leq 1/M$,

$$\begin{aligned}
\tilde{f}(t) &\leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla(f(x))\|_2^2 \\
&\leq f(x) - (t/2)\|\nabla f(x)\|_2^2 \\
&\leq f(x) - \alpha t\|\nabla f(x)\|_2^2,
\end{aligned}$$

since $\alpha < 1/2$. Therefore the backtracking line search terminates either with $t = 1$ or with a value $t \geq \beta/M$. This provides a lower bound on the decrease in the objective function. In the first case we have

$$f(x^+) \leq f(x) - \alpha\|\nabla f(x)\|_2^2,$$

and in the second case we have

$$f(x^+) \leq f(x) - (\beta\alpha/M)\|\nabla f(x)\|_2^2.$$

Putting these together, we always have

$$f(x^+) \leq f(x) - \min\{\alpha, \beta\alpha/M\}\|\nabla f(x)\|_2^2.$$

Now we can proceed exactly as in the case of exact line search. We subtract $p^\star$ from both sides to get

$$f(x^+) - p^\star \leq f(x) - p^\star - \min\{\alpha, \beta\alpha/M\}\|\nabla f(x)\|_2^2,$$

and combine this with $\|\nabla f(x)\|_2^2 \geq 2m(f(x) - p^\star)$ to obtain

$$f(x^+) - p^\star \leq (1 - \min\{2m\alpha, 2\beta\alpha m/M\})(f(x) - p^\star).$$

From this we conclude

$$f(x^{(k)}) - p^\star \leq c^k(f(x^{(0)}) - p^\star)$$

where

$$c = 1 - \min\{2m\alpha, 2\beta\alpha m/M\} < 1.$$

In particular, $f(x^{(k)})$ converges to $p^\star$ at least as fast as a geometric series with an exponent that depends (at least in part) on the condition number bound M/m. In the terminology of iterative methods, the convergence is at least linear.

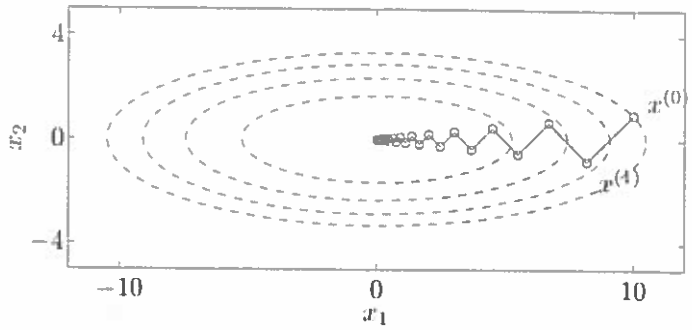

Figure 9.2 Some contour lines of the function $f(x) = (1/2)(x_1^2 + 10x_2^2)$. The condition number of the sublevel sets, which are ellipsoids, is exactly 10. The figure shows the iterates of the gradient method with exact line search, started at $x^{(0)} = (10, 1)$.

9.3.2 Examples

A quadratic problem in $\mathbf{R}^2$

Our first example is very simple. We consider the quadratic objective function on $\mathbf{R}^2$

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

where $\gamma > 0$. Clearly, the optimal point is $x^* = 0$, and the optimal value is 0. The Hessian of f is constant, and has eigenvalues 1 and γ, so the condition numbers of the sublevel sets of f are all exactly

$$\frac{\max\{1, \gamma\}}{\min\{1, \gamma\}} = \max\{\gamma, 1/\gamma\}.$$

The tightest choices for the strong convexity constants m and M are

$$m = \min\{1, \gamma\}, \qquad M = \max\{1, \gamma\}.$$

We apply the gradient descent method with exact line search, starting at the point $x^{(0)} = (\gamma, 1)$. In this case we can derive the following closed-form expressions for the iterates $x^{(k)}$ and their function values (exercise 9.6):

$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k,$$

and

$$f(x^{(k)}) = \frac{\gamma(\gamma + 1)}{2}\left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} = \left(\frac{\gamma - 1}{\gamma + 1}\right)^{2k} f(x^{(0)}).$$

This is illustrated in figure 9.2, for $\gamma = 10$.

For this simple example, convergence is exactly linear, *i.e.*, the error is exactly a geometric series, reduced by the factor $|(\gamma - 1)/(\gamma + 1)|^2$ at each iteration. For

$\gamma = 1$, the exact solution is found in one iteration; for γ not far from one (say, between 1/3 and 3) convergence is rapid. The convergence is very slow for $\gamma \gg 1$ or $\gamma \ll 1$.

We can compare the convergence with the bound derived above in §9.3.1. Using the least conservative values $m = \min\{1, \gamma\}$ and $M = \max\{1, \gamma\}$, the bound (9.18) guarantees that the error in each iteration is reduced at least by the factor $c = (1 - m/M)$. We have seen that the error is in fact reduced exactly by the factor

$$\left(\frac{1 - m/M}{1 + m/M}\right)^2$$

in each iteration. For small m/M, which corresponds to large condition number, the upper bound (9.19) implies that the number of iterations required to obtain a given level of accuracy grows at most like M/m. For this example, the exact number of iterations required grows approximately like $(M/m)/4$, *i.e.*, one quarter of the value of the bound. This shows that for this simple example, the bound on the number of iterations derived in our simple analysis is only about a factor of four conservative (using the least conservative values for m and M). In particular, the convergence rate (as well as its upper bound) is very dependent on the condition number of the sublevel sets.

A nonquadratic problem in $\mathbf{R}^2$

We now consider a nonquadratic example in $\mathbf{R}^2$, with

$$f(x_1, x_2) = e^{x_1 + 3x_2 - 0.1} + e^{x_1 - 3x_2 - 0.1} + e^{-x_1 - 0.1}. \tag{9.20}$$

We apply the gradient method with a backtracking line search, with $\alpha = 0.1$, $\beta = 0.7$. Figure 9.3 shows some level curves of f, and the iterates $x^{(k)}$ generated by the gradient method (shown as small circles). The lines connecting successive iterates show the scaled steps,

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

Figure 9.4 shows the error $f(x^{(k)}) - p^\star$ versus iteration k. The plot reveals that the error converges to zero approximately as a geometric series, *i.e.*, the convergence is approximately linear. In this example, the error is reduced from about 10 to about 10^{-7} in 20 iterations, so the error is reduced by a factor of approximately $10^{-8/20} \approx 0.4$ each iteration. This reasonably rapid convergence is predicted by our convergence analysis, since the sublevel sets of f are not too badly conditioned, which in turn means that M/m can be chosen as not too large.

To compare backtracking line search with an exact line search, we use the gradient method with an exact line search, on the same problem, and with the same starting point. The results are given in figures 9.5 and 9.4. Here too the convergence is approximately linear, about twice as fast as the gradient method with backtracking line search. With exact line search, the error is reduced by about 10^{-11} in 15 iterations, *i.e.*, a reduction by a factor of about $10^{-11/15} \approx 0.2$ per iteration.

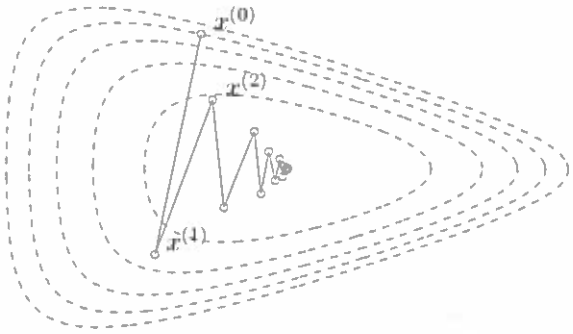

Figure 9.3 Iterates of the gradient method with backtracking line search, for the problem in $\mathbf{R}^2$ with objective f given in (9.20). The dashed curves are level curves of f, and the small circles are the iterates of the gradient method. The solid lines, which connect successive iterates, show the scaled steps $t^{(k)}\Delta x^{(k)}$.

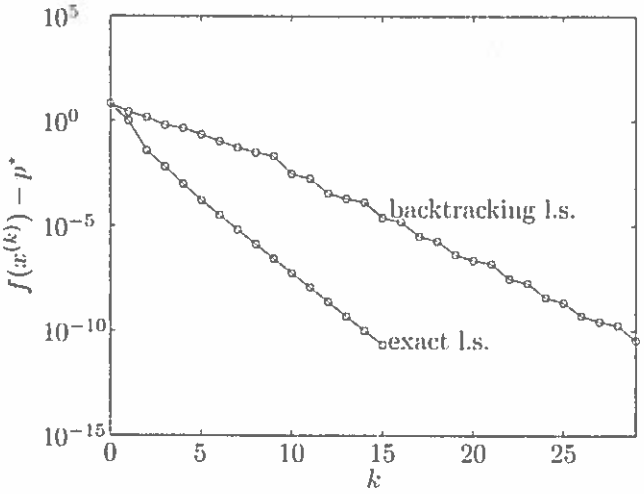

Figure 9.4 Error $f(x^{(k)}) - p^\star$ versus iteration k of the gradient method with backtracking and exact line search, for the problem in $\mathbf{R}^2$ with objective f given in (9.20). The plot shows nearly linear convergence, with the error reduced approximately by the factor 0.4 in each iteration of the gradient method with backtracking line search, and by the factor 0.2 in each iteration of the gradient method with exact line search.

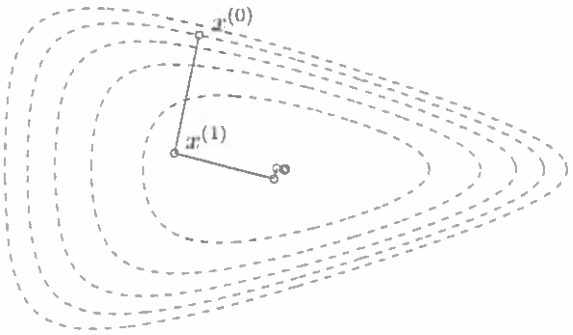

Figure 9.5 Iterates of the gradient method with exact line search for the problem in $\mathbf{R}^2$ with objective f given in (9.20).

A problem in $\mathbf{R}^{100}$

We next consider a larger example, of the form

$$f(x) = c^T x - \sum_{i=1}^{m} \log(b_i - a_i^T x), \tag{9.21}$$

with $m = 500$ terms and $n = 100$ variables.

The progress of the gradient method with backtracking line search, with parameters $\alpha = 0.1$, $\beta = 0.5$, is shown in figure 9.6. In this example we see an initial approximately linear and fairly rapid convergence for about 20 iterations, followed by a slower linear convergence. Overall, the error is reduced by a factor of around 10^6 in around 175 iterations, which gives an average error reduction by a factor of around $10^{-6/175} \approx 0.92$ per iteration. The initial convergence rate, for the first 20 iterations, is around a factor of 0.8 per iteration; the slower final convergence rate, after the first 20 iterations, is around a factor of 0.94 per iteration.

Figure 9.6 shows the convergence of the gradient method with exact line search. The convergence is again approximately linear, with an overall error reduction by approximately a factor $10^{-6/140} \approx 0.91$ per iteration. This is only a bit faster than the gradient method with backtracking line search.

Finally, we examine the influence of the backtracking line search parameters α and β on the convergence rate, by determining the number of iterations required to obtain $f(x^{(k)}) - p^\star \leq 10^{-5}$. In the first experiment, we fix $\beta = 0.5$, and vary α from 0.05 to 0.5. The number of iterations required varies from about 80, for larger values of α, in the range 0.2–0.5, to about 170 for smaller values of α. This, and other experiments, suggest that the gradient method works better with fairly large α, in the range 0.2–0.5.

Similarly, we can study the effect of the choice of β by fixing $\alpha = 0.1$ and varying β from 0.05 to 0.95. Again the variation in the total number of iterations is not large, ranging from around 80 (when $\beta \approx 0.5$) to around 200 (for β small, or near 1). This experiment, and others, suggest that $\beta \approx 0.5$ is a good choice.

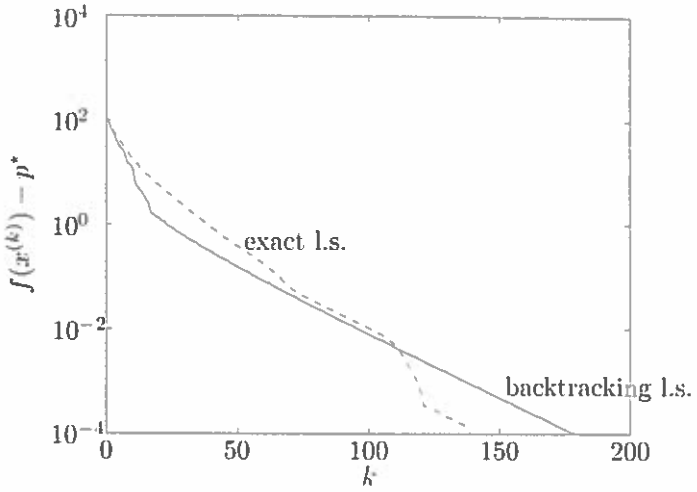

Figure 9.6 Error $f(x^{(k)}) - p^\star$ versus iteration k for the gradient method with backtracking and exact line search, for a problem in $\mathbf{R}^{100}$.

These experiments suggest that the effect of the backtracking parameters on the convergence is not large, no more than a factor of two or so.

Gradient method and condition number

Our last experiment will illustrate the importance of the condition number of $\nabla^2 f(x)$ (or the sublevel sets) on the rate of convergence of the gradient method. We start with the function given by (9.21), but replace the variable x by $x = T\bar{x}$, where

$$T = \mathbf{diag}((1, \gamma^{1/n}, \gamma^{2/n}, \ldots, \gamma^{(n-1)/n})),$$

i.e., we minimize

$$\bar{f}(\bar{x}) = c^T T\bar{x} - \sum_{i=1}^{m} \log(b_i - a_i^T T\bar{x}). \tag{9.22}$$

This gives us a family of optimization problems, indexed by γ, which affects the problem condition number.

Figure 9.7 shows the number of iterations required to achieve $\bar{f}(\bar{x}^{(k)}) - \bar{p}^\star < 10^{-5}$ as a function of γ, using a backtracking line search with $\alpha = 0.3$ and $\beta = 0.7$. This plot shows that for diagonal scaling as small as $10:1$ (*i.e.*, $\gamma = 10$), the number of iterations grows to more than a thousand; for a diagonal scaling of 20 or more, the gradient method slows to essentially useless.

The condition number of the Hessian $\nabla^2 \bar{f}(\bar{x}^\star)$ at the optimum is shown in figure 9.8. For large and small γ, the condition number increases roughly as $\max\{\gamma^2, 1/\gamma^2\}$, in a very similar way as the number of iterations depends on γ. This shows again that the relation between conditioning and convergence speed is a real phenomenon, and not just an artifact of our analysis.

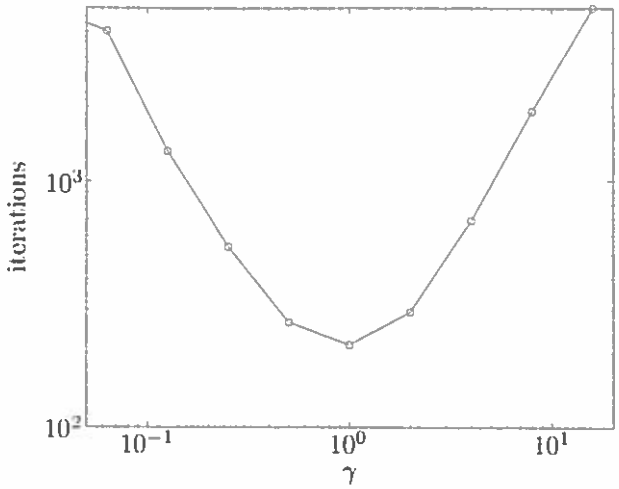

Figure 9.7 Number of iterations of the gradient method applied to problem (9.22). The vertical axis shows the number of iterations required to obtain $\bar{f}(\bar{x}^{(k)}) - \bar{p}^\star < 10^{-5}$. The horizontal axis shows γ, which is a parameter that controls the amount of diagonal scaling. We use a backtracking line search with $\alpha = 0.3$, $\beta = 0.7$.

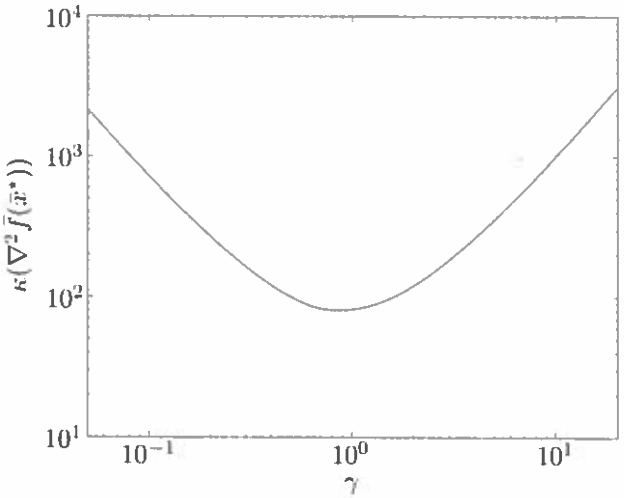

Figure 9.8 Condition number of the Hessian of the function at its minimum, as a function of γ. By comparing this plot with the one in figure 9.7, we see that the condition number has a very strong influence on convergence rate.

Conclusions

From the numerical examples shown, and others, we can make the conclusions summarized below.

- The gradient method often exhibits approximately linear convergence, *i.e.*, the error $f(x^{(k)}) - p^\star$ converges to zero approximately as a geometric series.
- The choice of backtracking parameters α, β has a noticeable but not dramatic effect on the convergence. An exact line search sometimes improves the convergence of the gradient method, but the effect is not large (and probably not worth the trouble of implementing the exact line search).
- The convergence rate depends greatly on the condition number of the Hessian, or the sublevel sets. Convergence can be very slow, even for problems that are moderately well conditioned (say, with condition number in the 100s). When the condition number is larger (say, 1000 or more) the gradient method is so slow that it is useless in practice.

The main advantage of the gradient method is its simplicity. Its main disadvantage is that its convergence rate depends so critically on the condition number of the Hessian or sublevel sets.

9.4 Steepest descent method

The first-order Taylor approximation of $f(x+v)$ around x is

$$f(x+v) \approx \widehat{f}(x+v) = f(x) + \nabla f(x)^T v.$$

The second term on the righthand side, $\nabla f(x)^T v$, is the *directional derivative* of f at x in the direction v. It gives the approximate change in f for a small step v. The step v is a descent direction if the directional derivative is negative.

We now address the question of how to choose v to make the directional derivative as negative as possible. Since the directional derivative $\nabla f(x)^T v$ is linear in v, it can be made as negative as we like by taking v large (provided v is a descent direction, *i.e.*, $\nabla f(x)^T v < 0$). To make the question sensible we have to limit the size of v, or normalize by the length of v.

Let $\|\cdot\|$ be any norm on $\mathbf{R}^n$. We define a *normalized steepest descent direction* (with respect to the norm $\|\cdot\|$) as

$$\Delta x_{\rm nsd} = \operatorname{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}. \tag{9.23}$$

(We say 'a' steepest descent direction because there can be multiple minimizers.) A normalized steepest descent direction $\Delta x_{\rm nsd}$ is a step of unit norm that gives the largest decrease in the linear approximation of f.

A normalized steepest descent direction can be interpreted geometrically as follows. We can just as well define $\Delta x_{\rm nsd}$ as

$$\Delta x_{\rm nsd} = \operatorname{argmin}\{\nabla f(x)^T v \mid \|v\| \leq 1\},$$