Chapter 3: Line Search Methods

Outline

Outline

General Description

- At each iteration, first computes a *search direction* p_k and then decides how far to move along that direction. The iteration is given by

$$x_{k+1} = x_k + \alpha_k p_k, \tag{1.1}$$

  where the positive scalar α_k is called the *step length*.
- The success of a line search method depends on effective choice of both the direction p_k and the step length α_k. In this chapter, we discuss
  - How to choose p_k and α_k to promote convergence from remote starting points;
  - Study the convergence results of several popular Line search algorithms.

Algorithm 1 (Line Search Method).

Input: Given initial point $x_0 \in \mathbb{R}^n$, the tolerance $\epsilon > 0$.

for $k = 0, 1, 2, \cdots$ **do**

If $\|g_k\| < \epsilon$, stop.

Find a descent direction p_k.

Find α_k such that $f(x_k + \alpha_k p_k) < f(x_k)$.

Set $x_{k+1} = x_k + \alpha_k p_k$.

end for

Outline

Taylor's Theorem

Theorem 1

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ *is a continuously differentiable and that* $p \in \mathbb{R}^n$. *Then we have*

$$f(x+p) = f(x) + \nabla f(x+tp)^T p, \tag{2.1}$$

for some $t \in (0,1)$. *Moreover, if* f *is twice continuously differentiable, we have that*

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp) p dt, \tag{2.2}$$

and that

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p, \tag{2.3}$$

for some $t \in (0,1)$.

Steepest Descent Direction for Line Search Methods

- Given any search direction p and step-length parameter α, we have

$$f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f_k + \frac{\alpha^2}{2} p^T \nabla^2 f(x_k + tp) p \tag{2.4}$$

  for some $t \in (0, \alpha)$.

- The rate of change in f along the direction p at x_k is $p^T \nabla f_k$. Hence, the unit direction p of most rapid decrease is the solution to the problem

$$\min_p p^T \nabla f_k, \text{ subject to } \|p\| = 1.$$

- Since $p^T \nabla f_k = \|p\| \|\nabla f_k\| \cos\theta = \|\nabla f_k\| \cos\theta$, where θ is the angle between p and ∇f_k, so the minimizer is attained when $\cos\theta = -1$ and

$$p = -\nabla f_k / \|\nabla f_k\|,$$

  as claimed, which is orthogonal to the contours of the function.

  Therefore, $-\nabla f_k$ is the one along which f decreases most rapidly.

Steepest Descent Direction for Line Search Methods

- The steepest descent direction $-\nabla f_k$ is the most obvious choice for search direction for a line search method.
- The line search method which moves along $p_k = -\nabla f_k$ at every step is called *steepest descent method*.
- One advantage of $-\nabla f_k$ is that it requires calculation of the gradient ∇f_k but not of second derivatives.
- However, it can be excruciatingly slow on difficult problems.

Search Directions for Line Search Methods

In general, any *descent* direction - one that satisfies $p^T \nabla f_k < 0$ - is guaranteed to produce a decrease in f, provided that the step length is sufficiently small. Actually, by using Taylor's theorem we have that

$$f(x_k + \epsilon p_k) = f(x_k) + \epsilon p_k^T \nabla f_k + O(\epsilon^2)$$

When p_k is a descent direction, it follows that $f(x_k + \epsilon p) < f(x_k)$ for sufficiently small positive values of ϵ.

Newton Direction for Line Search Methods

- Consider the second-order Taylor series approximation to $f(x_k + p)$, which is

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \equiv m_k(p). \tag{2.5}$$

- Assuming for the moment that $\nabla^2 f_k$ is positive definite, the *Newton direction* is obtained by finding the vector p that minimizes $m_k(p)$, namely

$$p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k. \tag{2.6}$$

Newton Direction for Line Search Methods

The Newton direction p^N is a descent direction when $\nabla f_k \neq 0$ and $\nabla^2 f_k$ is positive definite, because

$$\nabla f_k^T p_k^N = -(p_k^N)^T \nabla^2 f_k p_k^N \leq -\sigma_k \|p_k^N\|^2$$

for some $\sigma_k > 0$.

Newton Direction for Line Search Methods

- The Newton direction is reliable when the difference between the true function $f(x_k + p)$ and its quadratic model $m_k(p)$ is not too large.
- By comparing (2.5) with (2.3), we see that the only difference between these functions is that the matrix $\nabla^2 f(x_k + tp)$ in the third term of the expansion has been replaced by $\nabla^2 f$. If $\nabla^2 f$ is sufficiently smooth, this difference introduces a perturbation of only $O(\|p\|^3)$ into the expension, so that when $\|p\|$ is small, the approximation $f(x_k + p) \approx m_k(p)$ is quite accurate.
- Unlike the steepest descent direction, there is a "natural" step length of 1 associated with the Newton direction. Most line search implementations of Newton's method use the unit step $\alpha = 1$ where possible and adjust α only when it does not produce a satisfactory reduction in the value of f.

Newton Direction for Line Search Methods

- When $\nabla^2 f_k$ is not positive definite
  - $(\nabla^2 f_k)^{-1}$ may not exist, so p^N could not be well-defined.
  - Even it is defined, it may happen that $\nabla f_k^T p_k^N = -(p_k^N)^T \nabla^2 f_k p_k^N > 0$
- In this situations, line search methods modify the direction of p_k to make it satisfy the descent condition while retaining the benefit of the second-order information contained in $\nabla^2 f_k$. For instances choosing $\delta > 0$, define

$$p^N = -(\nabla^2 f_k + \delta I)^{-1} \nabla f_k.$$

Newton Direction for Line Search Methods

- Fast rate of local convergence, typically quadratic. After a neighborhood of the solution is reached, convergence to high accuracy often occurs in just a few iterations.
- Computation of the Hessian $\nabla^2 f(x)$. Explicit computation of this matrix of second derivatives can sometimes be a cumbersome, error prone and expensive process.
- Finite-difference and automatic differentiation techniques may be useful in avoiding the need to calculate second derivatives by hand.

Quasi-Newton Direction for Line Search Methods

- *Quasi-Newton* search directions provides an attractive alternative to Newton's method in that they do not require computation of the Hessian and yet still attain a superlinear rate of convergence.
- Use B_k to approximate the true Hessian $\nabla^2 f_k$. Define quasi-Newton direction

$$p_k = -B_k^{-1} \nabla f_k.$$

- Update B_k after each step to take account of the additional knowledge gained during the step. The updates make use of the fact that changes in the gradient g provide information about the second derivative of f along the search direction.

Quasi-Newton Direction for Line Search Methods

From $\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p dt$, we have

$$\nabla f(x+p) = \nabla f(x) + \nabla^2 f(x+p)p + \int_0^1 [\nabla^2 f(x+tp) - \nabla^2 f(x+p)]p dt.$$

Then

$$\nabla f_{k+1} = \nabla f_k + \nabla^2 f_{k+1}(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|).$$

When x_k and x_{k+1} lie in a region near the solution x^*, within which $\nabla^2 f$ is positive definite, the final term in this expansion is eventually dominated by the $\nabla^2 f_k(x_{k+1} - x_k)$ term, and we can write

$$\nabla^2 f_{k+1}(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k. \tag{2.7}$$

Quasi-Newton Direction for Line Search Methods

We choose the new Hessian approximation B_{k+1} so that it mimics the property (2.7) of the true Hessian, that is, we require it so satisfy the following condition, known as the *secant equation*:

$$B_{k+1} s_k = y_k, \tag{2.8}$$

where

$$s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k.$$

Typically, we impose additional conditions on B_{k+1}, such as symmetry (motivated by symmetry of the exact Hessian), and a requirement that the difference between successive approximations B_k and B_{k+1} have low rank.

Quasi-Newton Update Formulae

- *symmetric-rank-one* (SR1) formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}, \tag{2.9}$$

- *BFGS formula*, named after its inventors, Broyden, Fletcher, Goldfarb, and Shannon, which is defined by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{2.10}$$

- Both updates satisfy the secant equation and both maintain symmetry.
- One can show that BFGS update (2.10) generates positive definite approximations whenever the initial approximation B_0 is positive definite and $s_k^T y_k > 0$.

Search Directions for Line Search Methods

- One can obtain $p_k = -B_k^{-1}\nabla f_k$ by solving linear equations.
- Update B_k^{-1}, denoted as H_k. In fact, the equivalent formula for (2.9) and (2.10), applied to the inverse approximation $H_k \equiv B_k^{-1}$, is

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}, \tag{2.11}$$

and

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k s_k y_k^T) + \rho s_k s_k^T, \qquad p_k = \frac{1}{y_k^T s_k}. \tag{2.12}$$

- Calculation of p_k can then be performed by using the formula $p_k = -H_k \nabla f_k$. This matrix-vector multiplication is simpler than the factorization/back-substitution procedure.

Search Directions for Line Search Methods

- Most line search algorithms require p_k to be a *descent direction* - one for which $p_k^T \nabla f_k < 0$ - because this property guarantees that the function f can be reduced along this direction.
- Moreover, all the search directions we described above have the form

$$p_k = -B_k^{-1} \nabla f_k, \tag{2.13}$$

  where B_k is symmetric and nonsingular matrix.
- When B_k is positive definite, we have

$$p_k^T \nabla f_k = -\nabla^T f_k B_k^{-1} \nabla f_k < 0,$$

  and therefore p_k is a descent direction.

Search Directions for Line Search Methods

- In the steepest descent method, $B_k = I$;
- In Newton's method, $B_k = \nabla^2 f(x_k)$;
- In quasi-Newton methods, $B_k \approx \nabla^2 f(x_k)$ and is updated at every iteration by means of a low-rank formula.

Search Directions for Line Search Methods

The last class of search directions we preview here is that generated by *nonlinear conjugate gradient (CG) methods*. They have the form

$$p_k = -\nabla f(x_k) + \beta_k p_{k-1}, \qquad (2.14)$$

where β_k is a scalar that ensure that p_k and p_{k-1} are *conjugate* - an important concept in the minimization of quadratic functions.

Search Directions for Line Search Methods

- System of linear equations $Ax = b$, where the coefficient matric A is symmetric and positive definite.
- Equivalent to minimize the convex quadratic function

$$\phi(x) = \frac{1}{2}x^T Ax - b^T x,$$

  So it was natural to investigate extension of nonlinear CG methods to more general types of unconstrained minimization problems.
- In general, nonlinear conjugate directions are much more effective than the steepest descent direction and are almost simple to compute. These methods do not attain the fast convergence rates of Newton methods, but they have the advantage of not requiring storage of matrices.

Search Directions for Line Search Methods

- All of the search directions discussed so far can be used directly in a line search framework. They give rise to the steepest descent, Newton, quasi-Newton, and conjugate gradient line search methods.
- All except conjugate gradients have an analogue in the trust region framework.

Outline

Step Length

In computing the step length α_k, we face a tradeoff.

- We would like to choose α_k to give a substantial reduction of f,
- but at the same time we do not want to spend too much time making the choice.

Exact Line Search

- The ideal choice would be the global minimizer of the univariate function $\phi(\cdot)$ defined by

$$\phi(\alpha) = f(x_k + \alpha p_k), \qquad \alpha > 0, \tag{3.1}$$

  which leads to $p_k^T \nabla f(x_k + \alpha_k p_k) = 0$.
- But in general, it is too expensive to identify this value. To find even a local minimizer of ϕ to moderate precision generally requires too many evaluations of the objective function f and possibly the gradient ∇f.

Inexact Line Search

- Typical line search algorithms try out a sequence of candidate values for α, stopping to accept one of these values when certain conditions are satisfied.
- The line search is done in two stages:
  - *A bracketing phase* finds an interval containing desirable step lengths;
  - *A bisection or interpolation phase* computes a good step length within this interval.
- We now discuss various termination conditions for line search algorithms and show that effective step lengths need not lie near minimizers of the univariate function $\phi(\alpha)$ defined in (3.1).

A Simple Example

A simple condition we could impose on is to require in f, that is,

$$f(x_k + \alpha_k p_k) < f(x_k).$$

This requirement is not enough to produce convergence to x^*.

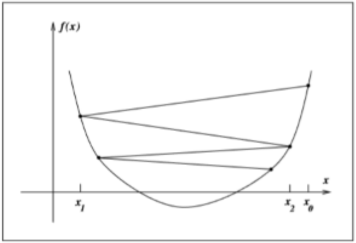


Figure 3.2 Insufficient reduction in f.

To avoid this behavior we need to enforce a *sufficient decrease condition*.

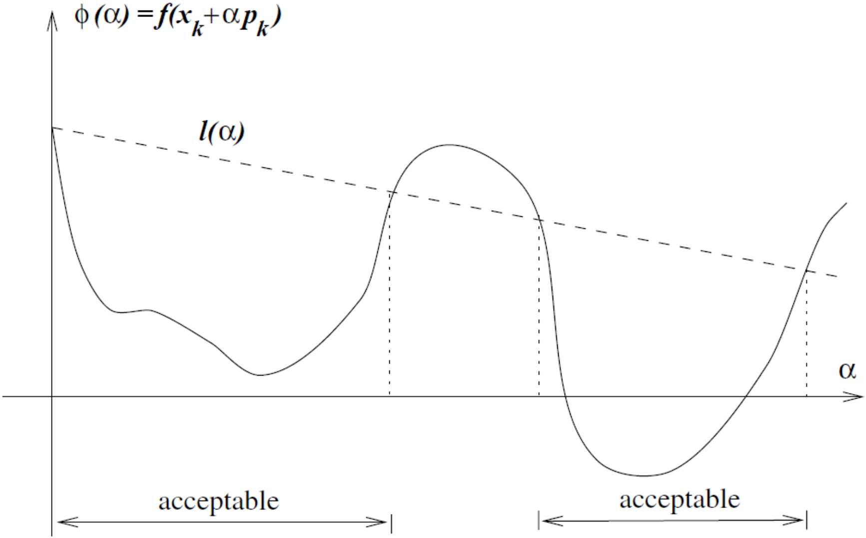
The Sufficient Decrease Condition

- Sometimes called as *Armijo condition*
- It stipulates that α_k should first of all give *sufficient decrease* in the objective function f:

$$f(x_k + \alpha p_k) \le f(x_k) + c_1 \alpha \nabla f_k^T p_k := l(\alpha), \tag{3.2}$$

  for some constant $c_1 \in (0, 1)$.
- In practice, c_1 is chosen to be quite small, say $c_1 = 10^{-4}$.
- (3.2) means that the reduction in f should be proportional to both the step length α_k and the directional derivative $\nabla f_k^T p_k$.

Sufficient Decrease Condition



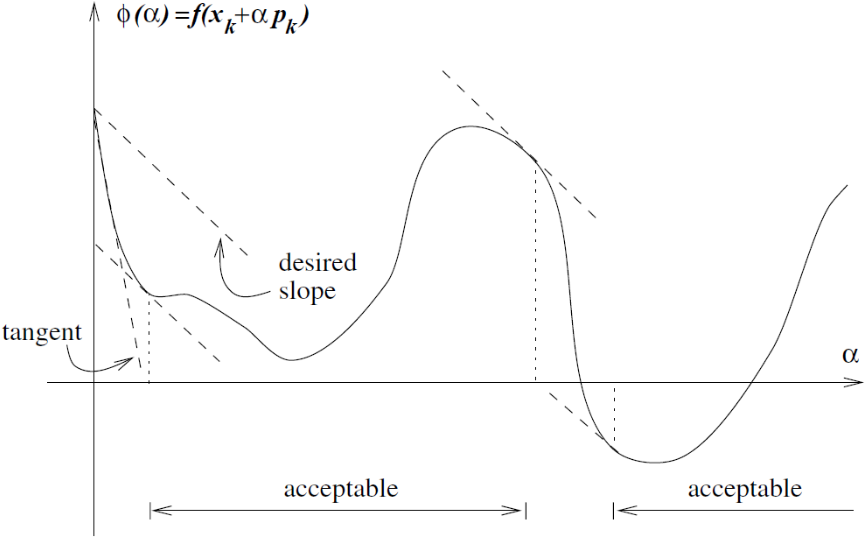
The Curvature Condition

- The sufficient decrease condition is not enough by itself to ensure that the algorithm makes reasonable progress because it is satisfied for all sufficiently small values of α.
- To rule out unacceptably short steps we introduce a second requirement, called the *curvature condition*, which requires α_k to satisfy

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \tag{3.3}$$

  for some constant $c_2 \in (c_1, 1)$, where c_1 is the constant from (3.2).
- Define $\phi(\alpha) = f(x_k + \alpha p_k)$, then (3.3) means $\phi'(\alpha_k) \geq c_2 \phi'(0)$

The Curvature Condition



The Curvature Condition

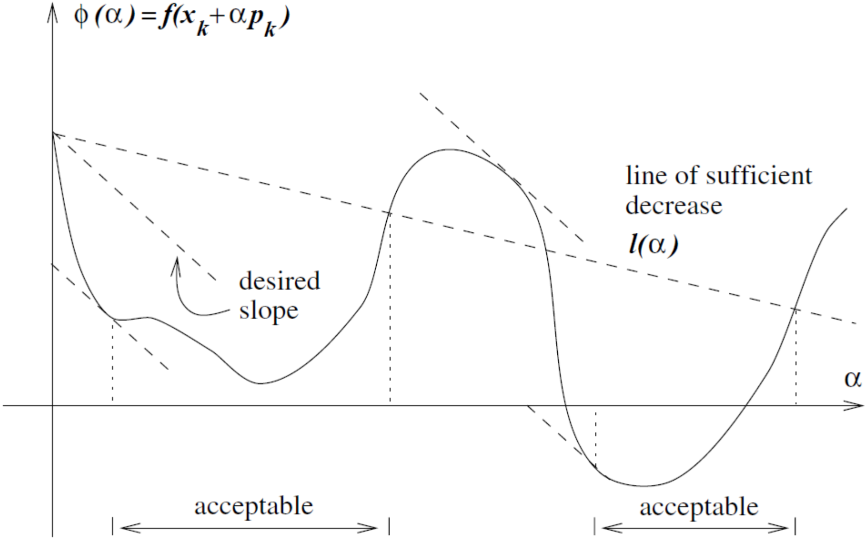
- $\phi'(\alpha_k) \geq c_2\phi'(0)$ ensures that the slope of ϕ at α_k is greater than c_2 times the initial step slope $\phi'(0)$. This make sense because if the slope $\phi'(\alpha)$ is strongly negatives, we have indication that we can reduce f significantly by moving further along the chosen direction.
- If $\phi'(\alpha_k)$ is only slightly negative or even positive, it is a sign that we cannot expect much more decrease in f in this direction, so it makes sense to terminate the line search.
- Typical values of c_2 are 0.9 when the search direction p_k is chosen by a Newton or quasi-Newton method, and 0.1 when p_k is obtained from a nonlinear CG method.

The Wolfe Conditions

$$
\begin{aligned}
f(x_k + \alpha_k p_k) &\leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k & \text{(3.4a)} \\
\nabla f(x_k + \alpha_k p_k)^T p_k &\geq c_2 \nabla f_k^T p_k, & \text{(3.4b)}
\end{aligned}
$$

with $0 < c_1 < c_2 < 1$. The Wolfe conditions are scale-invariant in a broad sense: Multiplying the objective function by a constant or making an affine change of variables does not alter them. They can be used in most line search methods, and are particularly important in the implementation of quasi-Newton methods.

The Wolfe Conditions



The Strong Wolfe Conditions

- A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of ϕ. We can, however, modify the curvature condition to force α_k to lie in at least a broad neighborhood of a local minimizer or stationary point of ϕ.
- The *strong Wolfe conditions* require α_k to satisfy

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \tag{3.5a}$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k|, \tag{3.5b}$$

  with $0 < c_1 < c_2 < 1$.
- Do not allow $\phi'(\alpha_k)$ to be too positive. Hence, we exclude points that are far from stationary points of ϕ.

The Wolfe Conditions

The following theorem shows that there exist step lengths that satisfy the Wolfe conditions for every function f that is smooth and bounded below.

Theorem 2

Suppose that $f : \Re^n \to \Re$ is continuously differentiable. Let p_k be a descent direction at x_k, and assume that f is bounded below along the ray $\{x_k + \alpha p_k | \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of step lengths satisfy the Wolfe conditions (3.4) and the strong Wolfe conditions (3.5).

Proof Sketch

Part 1. Note that $\phi(\alpha) = f(x_k + \alpha p_k)$ and the line $l(\alpha) = f(x_k) + \alpha c_1 \nabla f_k^T p_k$ must intersect at some points. Let $\alpha' > 0$ be the smallest intersecting value of α, that is

$$f(x_k + \alpha' p_k) = f(x_k) + \alpha' c_1 \nabla f_k^T p_k.$$

Part 2. There exists $\alpha'' \in (0, \alpha')$ such that

$$f(x_k + \alpha' p_k) - f(x_k) = \alpha' \nabla f(x_k + \alpha'' p_k)^T p_k.$$

Then $\nabla f(x_k + \alpha'' p_k)^T p_k = c_1 \nabla f_k^T p_k > c_2 \nabla f_k^T p_k$. Therefore, α'' satisfies both (3.4) and (3.5).

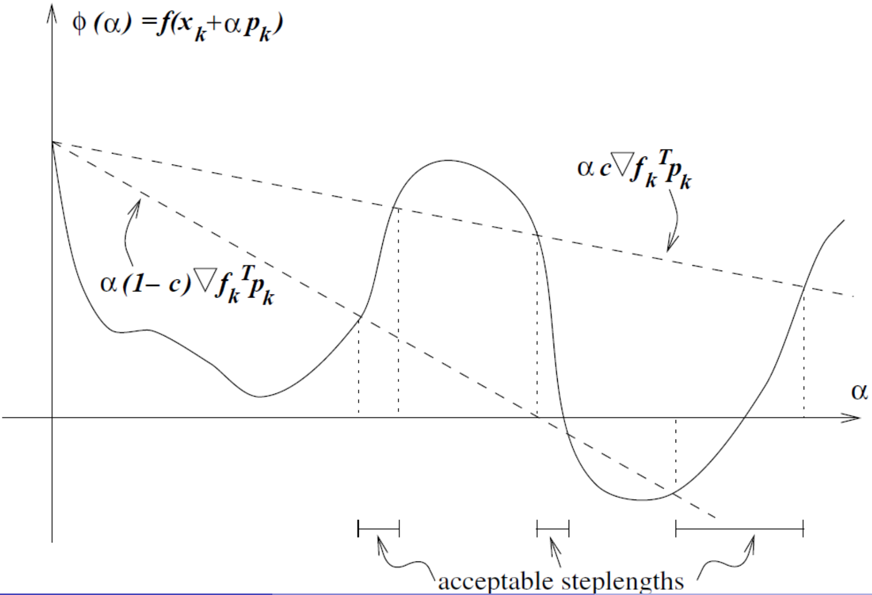
The Goldstein Conditions

The *Goldstein conditions* ensure that the step length α achieves sufficient decrease but is not too short:

$$f(x_k) + (1-c)\alpha_k \nabla f_k^T p_k \le f(x_k + \alpha_k p_k) \le f(x_k) + c\alpha_k \nabla f_k^T p_k, \tag{3.6}$$

with $0 < c < \frac{1}{2}$. The right inequality is the sufficient decrease condition (3.2), whereas the left inequality is introduced to control the step length from below.

The Goldstein Conditions



The Goldstein Conditions

- Disadvantage: the first inequality in (3.6) may exclude all minimizer of ϕ.
- However, the Goldstein and Wolfe conditions have much in common and their convergence theories are quite similar.
- The Goldstein conditions are often used in Newton-type methods but are not well suited for quasi-Newton methods, that maintain a positive definite Hessian approximation.

Sufficient Decrease and Backtracking

Algorithm: Backtracking Line Search.

Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;

repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$

$$\alpha \leftarrow \rho\alpha;$$

end(repeat)

Terminate with $\alpha_k = \alpha$.

Sufficient Decrease and Backtracking

- The initial step length $\bar{\alpha}$ is chosen to be 1 in Newton and quasi-Newton methods, but can have different values in other algorithms such as steepest descent or conjugate gradient.
- An acceptable step length will be found after a finite number of trials, because α_k will eventually become small enough that the sufficient decrease condition holds.
- In practice, the contraction factor ρ is often allowed to vary at each iteration of the line search. For example, it can be chosen by safeguarded interpolation. We need ensure only that at each iteration we have $\rho \in [\rho_{lo}, \rho_{hi}]$, for some fixed constants $0 < \rho_{lo} < \rho_{hi} < 1$.

Sufficient Decrease and Backtracking

- The backtracking approach ensures either that the selected step length α_k is some fixed value (the initial choice $\bar{\alpha}$), or else that it is short enough to satisfy the sufficient decrease condition but not *too* short.
- The latter claim holds because the accepted value α_k is within a factor ρ of the previous trial value, α_k/ρ, which was rejected for violating the sufficient decrease condition, that is, for being too long.
- Well suited for Newton methods but is less appropriate for quasi-Newton and conjugate gradient methods.

Step-Length Selection Algorithms

We now consider techniques for finding a minimum of the one-dimensional function

$$\phi(\alpha) = f(x_k + \alpha p_k), \tag{3.7}$$

or for simply finding a step length α_k satisfying one of the termination conditions we described.

Step-Length Selection Algorithms

- If f is a convex quadratic function $f(x) = \frac{1}{2}x^TQx - b^Tx$, its one-dimensional minimizer along the ray $x_k + \alpha p_k$ can be computed analytically and is given by

$$\alpha_k = -\frac{\nabla f_k^T p_k}{p_k Q p_k}.$$

- For general nonlinear functions, it is necessary to use an iterative procedure.

Step-Length Selection Algorithms

All the line search procedures require an initial estimate α_0 and generate a sequence α_i that either terminates with a step length satisfied by the user (for example, the Wolfe conditions) or determines that such a step length does not exist. Typical procedure consist of two phases:

- a *bracketing phase* that finds an interval $[\bar{a}, \bar{b}]$ containing acceptable step lengths, and
- a *selection phase* that zooms in to locate the final step length.

Interpolation

- The selection phase usually reduces the bracketing interval during its search for the desired length and interpolates some of the the function and derivative information gathered on earlier steps to guess the location of the minimizer.
- Rewrite the sufficient decrease condition in the notation of (3.7) as

$$\phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi(0) \tag{3.8}$$

Suppose that the initial guess α_0 is given. If we have

$$\phi(\alpha_0) \leq \phi(0) + c_1 \alpha_0 \phi(0), \tag{3.9}$$

this step length satisfies the condition, and we terminate the search. Otherwise, we know that the interval $[0, \alpha_0]$ contains acceptable step length.

Interpolation

We construct a quadratic approximation $\phi_q(\alpha)$ to ϕ so that it satisfies the interpolation conditions $\phi_q(0) = \phi(0)$, $\phi_q'(0) = \phi'(0)$, and $\phi_q(\alpha_0) = \phi(\alpha_0)$ as follow:

$$\phi_q(\alpha) = (\frac{\phi(\alpha_0) - \phi(0) - \alpha_0\phi'(0)}{\alpha_0^2})\alpha^2 + \phi'(0)\alpha + \phi(0).$$

The new trial value α_1 is defined as the minimizer of this quadratic, that is

$$\alpha_1 = -\frac{\phi'(0)\alpha_0^2}{2[\phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0]}.$$

Interpolation

If the sufficient decrease condition is satisfied at α_1, we terminate the search. Otherwise, we construct a *cubic* function that satisfies $\phi_c(0) = \phi(0)$, $\phi_c'(0) = \phi'(0)$, $\phi_c(\alpha_0) = \phi(\alpha_0)$ and $\phi_c(\alpha_1) = \phi(\alpha_1)$ as follow:

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \phi'(0)\alpha + \phi(0),$$

where

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\alpha_0^2\alpha_1^2(\alpha_1 - \alpha_0)} \begin{pmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{pmatrix} \begin{pmatrix} \phi(\alpha_1) - \phi(0) - \phi'(0)\alpha_1 \\ \phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0 \end{pmatrix}.$$

By differentiating $\phi_c(x)$, we see that the minimizer α_2 of ϕ_c lies in the interval $[0, \alpha_1]$ and is given by

$$\alpha_2 = \frac{-b + \sqrt{b^2 - 3a\phi'(0)}}{3a}.$$

Interpolation

- If necessary, above process is repeated, using a cubic interpolant of $\phi(0)$, $\phi'(0)$ and the two most recent values of ϕ, until an α that satisfies the sufficient decrease condition is located.
- If the computation of directional derivative can be done simultaneously with the function at little cost, we can design an alternative strategy based on cubic interpolation of the value of ϕ and ϕ' at the most recent values of α.
- Cubic interpolation provides a good model for functions with significant changes of curvature and usually produces a quadratic rate of convergence of the iteration to the minimizing value of α.

Initial Step Length

- For Newton and quasi-Newton methods the step $\alpha_0 = 1$ should always be used as the initial trial step length. This choice ensures that unit step lengths are taken whenever they satisfy the termination conditions and allows the rapid rate-of-convergence properties of these methods to take effect.
- For methods that do not produce well-scaled search directions, such as the steepest descent and conjugate gradient methods, it is important to use current information about the problem and the algorithm to make the initial guess.

Outline

Convergence of Line Search Methods

We discuss requirements on the search direction in this section, focusing on one key property: the angle θ_k between p_k and the steepest descent direction $-\nabla f_k$, defined by

$$\cos\theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|\|p_k\|}. \tag{4.1}$$

Convergence of Line Search Methods

Theorem 3 (Zoutendijk)

Consider any iteration of the form (1.1), where p_k is a descent direction and α_k satisfies the Wolfe conditions (3.4). Suppose that f is bounded below in $\Re^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} \equiv \{x : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on $\mathcal{N}$, that is, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(\widetilde{x})\| \leq L\|x - \widetilde{x}\|, \qquad \forall x, \widetilde{x} \in \mathcal{N}. \tag{4.2}$$

Then

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty, \tag{4.3}$$

which is called Zoutendijk condition.

Proof Sketch

Proof Sketch: Note that $x_{k+1} = x_k + \alpha_k p_k$. Lipschitz continuity of ∇f and the curvature condition

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k,$$

yields

$$(c_2 - 1)\nabla f_k^T p_k \leq (\nabla f_{k+1} - \nabla f_k)^T p_k \leq \alpha_k L \|p_k\|^2.$$

So we have $\alpha_k \geq \frac{(c_2 - 1)\nabla f_k^T p_k}{L\|p_k\|^2}$. Then sufficient decrease condition shows that

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \frac{c_1(1-c_2)}{L} \frac{(\nabla f_k^T p_k)^2}{\|p_k\|^2} \\ &\leq f(x_0) - \frac{c_1(1-c_2)}{L} \sum_{i=0}^{k} \cos^2 \theta_k \|\nabla f_k\|^2. \end{aligned}$$

Since f is lower bounded, we obtain (4.3).

Convergence of Line Search Methods

- Similar results to this theorem hold when the Goldstein condition or strong Wolfe conditions are used in place of the Wolfe conditions.
- The Zoutendijk condition (4.3) implies that

$$\cos^2 \theta_k \|\nabla f_k\|^2 \to 0. \tag{4.4}$$

This limit can be used in turn to derive global convergence results for line search algorithms.

Convergence of Line Search Methods

If our method for choosing the search direction p_k in the iteration (1.1) ensures that the angle θ_k defined by (4.1) is bounded away from 90°, there is a positive constant δ such that

$$\cos\theta_k \geq \delta > 0, \text{ for all } k. \tag{4.5}$$

It follows immediately from (4.4) that

$$\lim_{k\to\infty} \|\nabla f_k\| = 0. \tag{4.6}$$

In other words, we can be sure that the gradient norms $\|\nabla f_k\|$ converge to zero, provided that the search direction are never too close to orthogonality with the gradient.

Convergence of Line Search Methods

- The steepest descent method (for which $p_k = -\nabla f_k$, i.e. $\cos\theta_k = 1$) produces a gradient sequence that converges to zero
- Consider the Newton-like method $p_k = -B_k^{-1}\nabla f_k$ and assume that the matrices B_k are positive definite with a uniformly bounded condition number. That is, there is a constant M such that

$$\|B_k\|\|B_k^{-1}\| \leq M, \text{ for all } k.$$

It is easy to show from the definition (4.1) that

$$\cos\theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|\|p_k\|} = \frac{p_k^T B_k p_k}{\|p_k\|\|B_k p_k\|} \geq \frac{\lambda_{min}(B_k)}{\lambda_{max}(B_k)} \geq 1/M.$$

By combining this bound with (4.4) we find that

$$\lim_{k\to\infty} \|\nabla f_k\| = 0.$$

Convergence of Line Search Methods

- We use the term *globally convergent* to refer to algorithms for which the property (4.6) is satisfied.
- For line search methods of the general form (1.1), the limit (4.6) is the strongest global convergence result that can be obtained. We cannot guarantee that the method converges to a minimizer, but only that it is attracted by stationary points.
- Only by making additional requirements on the search direction p_k - by introducing negative curvature information from the Hessian $\nabla^2 f(x_k)$, for example - can we strengthen these results to include convergence to a local minimum.

Convergence of Line Search Methods

For some algorithms, such as conjugate gradient methods, we will not be able to prove the limit (4.6), but only the weaker result

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{4.7}$$

In other words, just a subsequence of the gradient norms $\|\nabla f_{k_j}\|$ converges to zero, rather than the whole sequence.

Convergence of Line Search Methods

In fact, we can prove global convergence in the sense of (4.6) or (4.7) for a general class of algorithms. Consider *any* algorithm for which

- every iteration produces a decrease in the objective function,
- every mth iteration is a steepest descent step, with step length chosen to satisfy the Wolfe or Goldstein conditions.

Then, since $\cos\theta_k = 1$ for the steepest descent steps, the result (4.7) holds.

The occasional steepest descent steps may not make much progress, but they at least guarantee overall global convergence.

Newton's Method with Hessian Modification

Algorithm 2 (Line Search Newton with Modification).

Given initial point x_0;

for $k = 0, 1, 2, \cdots$

Factorize the matrix $B_k = \nabla^2 f(x_k) + E_k$, where $E_k = 0$ if $\nabla^2 f(x_k)$ is sufficiently positive definite; otherwise, E_k is chosen to ensure that B_k is sufficiently positive definite;

Solve $B_k p_k = -\nabla f(x_k)$;

Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or Armijo backtracking conditions;

end

Newton's Method with Hessian Modification

Theorem 4

Let f be twice continuously differentiable on an open set $\mathcal{D}$, and assume that the starting point x_0 of Algorithm 2 is such the the level set $\mathcal{L} = \{x \in \mathcal{D} : f(x) \leq f(x_0)\}$ is compact,. Then if the bounded modified factorization property

$$\kappa(B_k) = \|B_k\| \, \|B_k^{-1}\| \leq C, \text{ for some } C > 0, \forall k = 0, 1, 2, \cdots$$

holds, we have that

$$\lim_{k \to \infty} \nabla f(x_k) = 0.$$

Outline

Rate of Convergence

- It would seem that designing optimization algorithms with good convergence properties is easy, since all we need to ensure is that the search direction p_k does not tend to become orthogonal to the gradient ∇f_k, or that steepest descent steps are taken regularly. However, it may lead to low convergence rate in practice, such as the steepest descent method.
- Algorithmic strategies that achieve rapid convergence can sometimes conflict with the requirements of global convergence. For example, the pure Newton iteration converges rapidly when started close enough to a solution, but its steps may not even be descent directions away from the solution.
- The challenge is to design algorithms that incorporate both properties: good global convergence guarantees and a rapid rate of convergence.

Convergence Rate of Steepest Descent

Theorem 5

Suppose that $f : \Re^n \to \Re$ is twice continuously differentiable, and that the iterates generated by the steepest-descent method with exact line searches converges to a point x^* at which the Hessian matrix $\nabla^2 f(x^*)$ is positive definite. Let r be any scalar satisfying

$$r \in (\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1).$$

where $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are the eigenvalues of $\nabla^2 f(x^*)$. Then for all k sufficiently large we have

$$f(x_{k+1}) - f(x^*) \le r^2 [f(x_k) - f(x^*)].$$

- Detailed proof is referred to Theorem 3.4 in [1].

Convergence Rate of Steepest Descent

- In general, we cannot expect the rate of convergence to improve if an inexact line search is used.
- The steepest descent method can give an unacceptable slow rate of convergence, even when the Hessian is reasonably well conditioned.
- For example, if condition number $\kappa(Q) = \lambda_n/\lambda_1 = 800$, $f(x_1) = 1$ and $f(x^*) = 0$, the above theorem suggest that the function value will still be about 0.08 after one thousand iterations of the steepest decent method with exact line search.

Newton's Method

Theorem 6

Suppose that f is twice differentiable and that the Hessian $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of a solution x^* at which the second-order sufficient conditions are satisfied. Consider the iteration $x_{k+1} = x_k + p_k$, where

$$p_k^N = -\nabla^2 f_k^{-1} \nabla f_k.$$

Then

(i) if the starting point x_0 is sufficiently close to x^*, the sequence of iterates converges to x^*;

(ii) the rate of convergence of $\{x_k\}$ is quadratic; and

(iii) the sequence of gradient norms $\{\|\nabla f_k\|\}$ converges quadratically to zero.

Detailed proof is referred to Theorem 3.5 in [1].

Quasi-Newton Methods

Theorem 7

Suppose that $f : \Re^n \to \Re$ *is twice continuously differentiable. Consider the iteration* $x_{k+1} = x_k + p_k$ *and that* p_k *is given by*

$$p_k = -B_k^{-1} \nabla f_k,$$

where the symmetric and positive definite matrix B_k *is updated at every iteration by a quasi-Newton updating formula. Let us assume that* $\{x_k\}$ *converges to a point* x^* *such that* $\nabla f(x^*) = 0$ *and* $\nabla^2 f(x^*)$ *is positive definite. Then* $\{x_k\}$ *converges superlinearly if and only if*

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0 \tag{5.1}$$

holds.

Remark: (5.1) is equivalent to $\lim_{k\to\infty} \frac{\|\nabla f_k + \nabla^2 f(x_k) p_k\|}{\|p_k\|} = 0$. Detailed proof is referred to Theorem 3.7 in [1].

Outline

Summary

Algorithmic strategies that achieve rapid convergence can sometimes conflict with the requirements of global convergence, and vice versa. For example

- the steepest descent method is the quintessential global convergent algorithm, but it is quite slow in practice.
- the pure Newton iteration converges rapidly when started close enough to a solution, but its steps may not even be descent directions away from the solution.

The challenge is to design algorithms that incorporate both properties: good global convergence guarantees and a rapid rate of convergence.

References

Jorge Nocedal and Stephen J. Wright, Numerical Optimization, second edition, Springer, 2006.

Thanks for your attention!