Review for Nonlinear Programming

Zhiting Xu

November 30, 2008

1 Line Search Methods

In line search method, each iteration computes a search direction p_k and then decides how far to move along that direction. That is,

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

The search direction p_k often has the form

$$p_k = -B_k^{-1} f_k \tag{1.2}$$

where B_k is a symmetric and nonsingular matrix.

1.1 Step Length

Define $\phi(\alpha) = f(x_k + \alpha p_k), \alpha > 0$. The ideal choice of α would be global minimizer of ϕ. However, it is usually too expensive to identify this value. Line search methods try to find a reasonable value α that meets some conditions by try out a sequence of candidate α. A popular condition is *Wolfe Conditions*, which has two inequality:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k \tag{1.3}$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k \tag{1.4}$$

where $0 < c_1 < c_2 < 1$.

The right-hand-side of 1.3 is a linear function with negative slop, so the first Wolfe Condition is that the function value of the new point should be sufficient small.

To avoid a too short step, the Wolfe Condition also requires a smooth slope, which is the second Wolfe Condition.

The *Strong Wolfe conditions* requires that the derivative $\phi'(\alpha_k)$ can't be too positive:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \tag{1.5a}$$

$$|\nabla f(x_k + \alpha_k f_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \tag{1.5b}$$

1.2 Convergence of Line Search

To see the convergence of line search, discuss the angle θ_k between p_k and the steepest descent direction, define by

$$\cos\theta_k = \frac{-\nabla f_k^T p_k}{||\nabla f_k||||p_k||} \tag{1.6}$$

Theorem 1.1 *Consider any iteration of the form 1.1, where p_k is a descent direction and α_k satisfies the Wolfe conditions 1.3, 1.4. Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} \stackrel{def}{=} \{x : f(x) \leq f(x_0)\}$,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*

$$||\nabla f(x) - \nabla f(\tilde{x})|| \leq L||x - \tilde{x}||, \text{for all } x, \tilde{x} \in \mathcal{N} \tag{1.7}$$

Then

$$\sum_{k \geq 0} \cos^2 \theta_k ||\nabla f_k||^2 < \infty \tag{1.8}$$

If the angle between p_k and $-\nabla f_k$ is bounded away from 90^0, that is, $\cos\theta_k \geq \delta > 0$, for all k, then $\lim_{k\to\infty} ||\nabla f_k|| = 0$. If Newton and quasi-Newton methods require B_k bounded, that is, $||B_k||||B_k^{-1}|| \leq M$, then $\cos\theta_k \geq 1/M$. Therefore, these methods are globally convergent.

For conjugate gradient methods, we can only get weaker result: $\liminf_{k\to\infty} ||\nabla f_k|| = 0$.

1.3 Rate of Convergence

Basic concepts:

$$\text{Q-linear:} \quad \frac{||x_{k+1} - x^*||}{||x_k - x^*||} \leq r, \text{for all k sufficiently large} \tag{1.9}$$

$$\text{Q-superlinear:} \quad \lim_{k\to\infty} \frac{||x_{k+1} - x^*||}{||x_k - x^*||} = 0 \tag{1.10}$$

$$\text{Q-quadratic:} \quad \frac{||x_{k+1} - x^*||}{||x_k - x^*||^2} \leq M \tag{1.11}$$

R-linear: if there is a sequence of nonnegative scalars $\{v_k\}$ such that $||x_k - x^*|| \leq v_k$ for all k, and $\{x_k\}$ converges Q-linearly to zero. The sequence $||x_k - x^*||$ is said to be dominated by $\{v_k\}$.

Steepest Descent

Theorem 1.2 *When the steepest descent method with exact line searches $x_{k+1} = x_k - \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k} \nabla f_k$ is applied to the strongly convex quadratic function $f(x) = \frac{1}{2}x^T Q x - b^T x$, the error norm $\frac{1}{2}||x - x^*||_Q^2 = f(x) - f(x)^*$ satisfies*

$$||x_{k+1} - x^*||_Q^2 \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}^2 ||x_k - x^*||_Q^2 \tag{1.12}$$

where $0 < \lambda_1 \leq \lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvalues of Q.

Newton's Method

In Newton iteration, the search p_k is given by:

$$p_k^N = -\nabla^2 f_k^{-1} \nabla f_k \tag{1.13}$$

p_k may not be the descent direction as $\nabla^2 f_k$ may not be positive definite.

Theorem 1.3 *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 sufficient conditions are satisfied. Consider the iteration $x_{k+1} = x_k + p_k$, where p_k is given by 1.13. Then*

- *1- if the starting point x_0 is sufficiently close to $x*$, the sequence of iterates converges to x^**
- *2- the rate of convergence of $\{x_k\}$ is quadratic; and*
- *3- the sequence of gradient norms $\{||\nabla f_k||\}$ converges quadratically to zero.*

Quasi-Newton Methods

In Quasi-Newton method, p_k is:

$$p_k = -B_k^{-1} \nabla f_k \tag{1.14}$$

where B_k is symmetric and positive definite.

Theorem 1.4 *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Consider the iteration $x_{k+1} = x_k + a_k p_k$, where p_k is a descent direction and α_k satisfies the Wolfe conditions 1.3, 1.4 with $c_1 \leq 1/2$. If the sequence $\{x_k\}$ converges to a point x^* such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, and if the search direction satisfies*

$$\lim_{k\to\infty} \frac{||\nabla f_k + \nabla^2 f_k p_k||}{||p_k||} = 0 \tag{1.15}$$

then

- *1- the step length α_k is admissible for all k greater than a certain index k_0: and*
- *2- if $\alpha_k = 1$ for all $k > k_0$,$\{x_k\}$ converges to x^* superlinearly.*

If p_k is a quasi-Newton search direction of the from 1.14, then 1.15 is equivalent to

$$\lim_{k\to\infty} \frac{||(B_k - \nabla^2 f(x^*))p_k||}{||p_k||} = 0 \tag{1.16}$$

This is *both necessary and sufficient* for the superlinear convergence of quasi-Newton methods.

Theorem 1.5 *Suppose that $f : \mathbb{R}^n \to R$ is twice continuously differentiable. Consider the iteration $x_{k+1} = x_k + p_k$(that is, the step length α_k is uniformly 1) and that p_k is given by 1.14. Let us assume also 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 1.16 holds.*

1.4 Step-Length Selection Algorithms

Line search uses an initial estimate α_0 and generates a sequence $\{\alpha_i\}$ that either terminates at a step that meets some conditions (like Wolfe condition), or determines that such a step length does not exist. Basically, the precedure consists of two phases: *a bracketing phase* that finds an interval $[a, b]$ that contains acceptable step lengths, and then a *selection phase* taht zooms in to locate the final step length. The selection phase usually reduces the bracketing interval and interpolates some of the function and derivative information gathered on earlier steps to guess the location of the minimizer.

Interpolation

At a guess α_i, if we have

$$\phi(\alpha_i) \leq \phi(0) + c_1 \alpha_i \phi'(0) \tag{1.17}$$

Then this step length satisfies the condition. Otherwise, we know that $[0, \alpha_i]$ contains acceptable step lengths. Perform a quadratic approximation $\phi_q(\alpha)$ to ϕ by interpolating the three pieces of information available - $\phi(0), \phi'(0)$, and $\phi(\alpha_i)$- to obtain

$$\phi_q(\alpha) = (\frac{\phi(\alpha_i) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2}) + \phi'(0)\alpha + \phi(0) \tag{1.18}$$

Initial Step Length

For Newton and quasi-Newton methods, the step $\alpha_0 = 1$ should always be used as the initial trial step length. For methods that do not produce well scaled search directions, such as the steepest descent and conjugate gradient methods, use current information about the problem and the algorithm to make the initial guess.

1.5 Barzilai-Borwein

$$\begin{aligned} s_k &= x_k - x_{k-1} \\ y_k &= \nabla f(x_k) - \nabla f(x_{k-1}) \end{aligned}$$

In Newton method, $p_k = -\nabla f^2(x_k)\nabla f(x_k)$.From Taylor theorem, we have

$$\nabla f^2(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1}) \tag{1.19}$$

,which is secant condition.

Algorithm 1 Line Search Algorithm

Set $\alpha_0 \leftarrow 0$, choose $\alpha_{max} > 0$ and $\alpha_1 \in (0, \alpha_{max})$
$i \leftarrow 1$

repeat
  Evaluate $\phi(\alpha_i)$;
  if $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)or[\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ and $i > 1]$ **then**
    $\alpha_* \leftarrow$ **zoom(**α_{i-1}, α_i**)** and **stop**;
  end if
  Evaluate $\phi'(\alpha_i)$
  if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$ **then**
    set $\alpha_* \leftarrow \alpha_i$ and stop
  end if
  if $\phi'(\alpha_i) \geq 0$ **then**
    set $\alpha_* \leftarrow zoom(\alpha_i, \alpha_{i-1})$ and stop
  end if
  Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{max})$
  $i \leftarrow i + 1$
until

Algorithm 2 zoom

repeat
  Interpolate to find a trial step length α_j between α_{lo} and α_{hi}
  Evaluate $\phi(\alpha_j)$
  if $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$or $\phi(\alpha_j) \geq \phi(\alpha_{lo})$ **then**
    $\alpha_{hi} \leftarrow \alpha_j$
  else
    Evaluate $\phi'(\alpha_j)$
    if $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$ **then**
      Set $\alpha_* \leftarrow \alpha_j$ and stop
    end if
    if $\phi'(\alpha_j)(\alpha_{hi} - \alpha_{lo}) \geq 0$ **then**
      $\alpha_{hi} \leftarrow \alpha_{lo}$
    end if
    $\alpha_{lo} \leftarrow \alpha_j$
  end if
until

In quasi Newton method, use B instead of H.

In Barzilai-Borwein, use $B_k = \alpha_k I$, choose $\alpha_k > 0$ that $B_k s_k \approx y_k$, that is, $\alpha s \approx y$.

$$\begin{aligned}
&\min_{\alpha} ||\alpha s - y||_2^2 \\
&\min_{\alpha} (\alpha s - y)^T(\alpha s - y) \\
&\alpha = \frac{s^T y}{s^T s}
\end{aligned}$$

Then we have

$$\begin{aligned}
&\alpha_k p_k = -\nabla f(x_k) \\
&p_k = -\frac{1}{\alpha_k}\nabla f(x_k) \\
&x_{k+1} = x_k - \frac{s_k^T s_k}{s_k^T y_k}\nabla f(x_k) \qquad (1.20)
\end{aligned}$$

Alternative BB formula
Try to approximate $\nabla^2 f(x_k)^{-1}$ rather than $\nabla^2 f(x_k)$
State secant condition as: $s_k \approx f(x_k)^{-1} y_k$. Let $\tau_k I = \nabla^2 f(x_k)^{-1}$, so

$$\begin{aligned}
\tau_k &= \text{argmin}||s_k - \tau_k y_k||_2^2 \\
&= \frac{s_k^T y_k}{y_k^T y_k} \qquad (1.21)
\end{aligned}$$

Switched BB
Take BB step when k is even, and take BBalt step when k is odd.
Cyclic BB
Choose cycle length M, recompute α_k using B every Mth iteration.
Usually, cycle BB performs better than other BB methods.

2 Trust-Region Methods

Trust-region methods define a region around the current iterate within which they *trust* the model to be an adequate representation of the objective function, and then choose the step to be the approximate minimizer of the model in this region. They choose the direction and length of the step simultaneously.

Taylor-series expansion of f around x_k:

$$f(x_k + p) = f_k + g_k^T p + \frac{1}{2} p^T \nabla^2 f(x_k + tp) p \tag{2.1}$$

By using an approximation B_k to the Hessian in the second-order term, m_k is defined as:

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \tag{2.2}$$

To obtain each step, we seek a solution of the subproblem

$$\min_{p \in R^n} m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \text{ s.t.} ||p|| \leq \Delta_k \tag{2.3}$$

Measure agreement between p_k and f using ratios ρ of actual of two protected decrease:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} = \frac{actual}{protected} \tag{2.4}$$

Solving the TR subproblem

$$\min_{p \in R^n} m(p) = f + g^T p + \frac{1}{2} p^T B p \text{ s.t.} ||p|| \leq \Delta \tag{2.5}$$

Theorem 2.1 *The vector p^* is a global solution of the trust-region problem*

$$\min_{p \in R^n} m(p) = f + g^T p + \frac{1}{2} p^T B p \ s.t. ||p|| \leq \Delta \tag{2.6}$$

if and only if p^* is feasible and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:

$$(B + \lambda I) p^* = -g \tag{2.7a}$$
$$\lambda(\Delta - ||p^*||) = 0 \tag{2.7b}$$
$$(B + \lambda I) \textit{is positive semidefinite} \tag{2.7c}$$

Algorithm 3 Trust Region

given $\hat{\Delta} > 0, \Delta_0 \in (0, \hat{\Delta})$, and $\eta \in [0, 1/4]$
for $k = 0, 1, 2$ **do**
  Obtain p_k by (approximately) solving 2.5
  Evaluate ρ_k from 2.4
  if $\rho < 1/4$ **then**
    $\Delta_{k+1} = 1/4\Delta_k$
  else
    if $\rho_k > 3/4$ and $||p_k|| = \Delta_k$ **then**
      $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$
    else
      $\Delta_{k+1} = \Delta_k$
    end if
  end if
  if $\rho_k > \eta$ **then**
    $x_{k+1} = x_k + p_k$
  else
    $x_{k+1} = x_k$
  end if
end for

2.1 Algorithms based on the Cauchy point

The Cauchy point

Algorithm(Cauchy Point Calculation)

Find the vector p_k^s taht solves a linear version of 2.5, that is,

$$p_k^s = \min_{p \in R^n} f_k + g_k^T p \text{ s.t.} ||p|| \leq \Delta_k \tag{2.8}$$

Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^s)$ subject to satisfying the trust-region bound, that is,

$$\tau_k = \min_{\tau \geq 0} m_k(\tau p_k^s) \text{ s.t.} ||\tau p_k^s|| \leq \Delta_k \tag{2.9}$$

Set $p_k^c = \tau_k p_k^s$

The solution of 2.8 is

$$p_k^s = -\frac{\Delta_k}{||g_k||} g_k \tag{2.10}$$

If $g_k^T B g_k \leq 0$, τ_k is 1. Otherwise, τ_k is minimizer of $||g||^3/(\Delta_k g_k^T B_k g_k)$. In summary, $p_k^c = -\tau \frac{\Delta_k}{||g_k||} g_k$, where $\tau_k = 1$ if $g_k^T B_k g_k \leq 0$ or $\min(||g_k||^3/(\Delta_k g_k^T B_k g_k), 1)$ otherwise.

The Dogleg Method

It can be used when B is positive definite. When B is positive definite, the unconstrained minimize of m of problem 2.6 is $p^B = -B^{-1}g$. When this point is feasible for 2.6, it is a solution. When Δ is small relative to p^B, the restriction $p \leq \Delta$ ensures that quadratic term in m has little effect on the solution of 2.6. Then omit the quadratic term, the solution is $p \approx -\Delta \frac{g}{||g||}$

The dobleg method finds a path consisting of two line segments. The first one runs along the steepest descent direction, which is

$$p^U = -\frac{g^T g}{g^T B g} g \tag{2.11}$$

while the second one runs from p^U to p^B. That is

$$\tilde{p}(\tau) = \begin{cases} \tau p^U & : \quad 0 \leq \tau \leq 1 \\ p^U + (\tau - 1)(p^B - p^U) & : \quad 1 \leq \tau \leq 2 \end{cases}$$

Lemma 2.1 *Let B be positive definte. Then*

- $||\tilde{p}(\tau)||$ *is an increasing function of* τ, *and*
- $m(\tilde{p}(\tau))$ *is a decreasing function of* τ

2.2 Global Convergence

The dogleg produces approximate solutions p_k satisfies:

$$m_k(0) - m_k(p_k) \geq c_1 ||g_k|| \min(\Delta_k, \frac{||g_k||}{||B_k||}) \tag{2.12}$$

Lemma 2.2 *The Cauchy point p_k^c satisfies 2.12 with $c_1 = 1/2$.*

Theorem 2.2 *Let $\eta = 0$ in Algorithm 2.1. Suppose that $||B_k|| \leq \beta$ for some constant β, that f is bounded below on the level set $S = \{x | f(x) \leq f(x_0)\}$ and Lipschitz continuously differentiable in the neighborhood $S(R_0)$ for some $R_0 > 0$, and that all approximate solutions of 2.5 satisfy the inequalities 2.12 and $||p_k|| \leq$
gammaΔ_k, for some positive constants c_1 and γ. We then have*

$$\liminf_{k \to \infty} ||g_k|| = 0 \tag{2.13}$$

Theorem 2.3 *$\eta > 0$, accept p_k only if $\rho_k > eta$, i.e. $f(x_k) - f(x_k + p_k) \geq \eta(m_k(0) - m_k)(p_k)$. Then $\lim_{k \to \infty} ||g|| = 0$*

2.3 Iterative solution of the subproblem

Try harder to solve the subproblem. Use theorem 2.1 with an eigenvalue decomposition to get an explicit formula for p.

$$p(\lambda) = -Q(\Lambda + \lambda I)^{-1} Q^T g = -\sum_{j=1}^{n} \frac{q_j^T g}{\lambda_j + \lambda} q_j \tag{2.14}$$

By orthonormality of $q_1, q_2, \dots, q_n$, we have

$$||p(\lambda)|| = \sum_{j=1}^{n} \frac{(q_j^T g)^2}{(\lambda_j + \lambda)^2} \tag{2.15}$$

Do line search to find $\lambda > -\lambda_1$ such that $||p(\lambda))^2 - \Delta^2 = 0$.
TR with true Hessian works even when Hessian is not positive definite.

3 Conjugate Gradient Methods

3.1 The linear conjugate gradient method

The conjugate gradient method is an iterative method for solving a linear system of equations

$$Ax = b \tag{3.1}$$

where A is an $n \times n$ symmetric positive definite matrix. It is equal to

$$\min \phi(x) = \frac{1}{2}x^T Ax - b^T x \tag{3.2}$$

Its gradient equals to the residual of the linear system

$$\nabla\phi(x) = Ax - b = r(x) \tag{3.3}$$

A set of nonzero vectors $\{p_0, p_1, \dots, p_l\}$ is said to be conjugate with respect to the symmetric positive definite matrix A if

$$p_i^T A p_j = 0 \tag{3.4}$$

We can minimize ϕ in n steps if we minimize it along the individual directions in a conjugate set. That is

$$x_{k+1} = x_k + \alpha_k p_k \tag{3.5}$$

where α_k is the one-dimensional minimizer of the quadratic function ϕ along $x_k + \alpha p_k$

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} \tag{3.6}$$

Theorem 3.1 *For any $x_0 \in R^n$ the sequence $\{x_k\}$ generated by the conjugate direction algorithm 3.5, 3.6 converges to the solution x^* of the linear system 3.1 in at most n steps.*

Conjugate Directions

From 3.3 and 3.5, we have

$$r_{k+1} = r_k + \alpha_k A p_k \tag{3.7}$$

Theorem 3.2 *Let $x_0 \in R^n$ be any starting point and suppose that the sequence $\{x_k\}$ is generated by the conjugate direction algorithm 3.5, 3.6. Then*

$$r_k^T p_i = 0 fori = 0, 1, \dots, k-1 \tag{3.8}$$

and x_k is the minimizer of $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ over the set

$$\{x | x = x_0 + span\{p_0, p_1, \dots, p_{k-1}\}\} \tag{3.9}$$

Basic properties of the conjugate gradient method

The conjugate gradient method is a conjugate direction method with very special property: in generating its set of conjugate vectors, it can compute a new vector p_k by using only the previous vector p_{k-1}.

$$p_k = -r_k + \beta_k p_{k-1} \tag{3.10}$$

As $p_{k-1}^T A p_k = 0$,

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}} \tag{3.11}$$

The residuals r_i are mutually orthogonal. Each search direction p_k and residual r_k is contained in the *Krylov subspace of degree k for r_0*, defined as

$$\mathcal{K}(r_0; k) = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\} \tag{3.12}$$

Theorem 3.3 *Suppose that the kth iterate generated by the conjugate gradient method is not the solution point x^*. The following four properties hold:*

$$r_k^T r_i = 0 \textit{ for } i = 0, 1, \ldots, k-1 \tag{3.13a}$$

$$\textit{span}\{r_0, r_1, \ldots, r_k\} = \textit{span}\{r_0, Ar_0, \ldots, A^k r_0\} \tag{3.13b}$$

$$\textit{span}\{p_0, p_1, \ldots, p_k\} = \textit{span}\{r_0, Ar_0, \ldots, A^k r_0\} \tag{3.13c}$$

$$p_k^T A p_i = 0, \textit{ for } i = 0, 1, \ldots, k-1 \tag{3.13d}$$

Therefore, the sequence $\{x_k\}$ converges to x^* in at most n steps.

A practical form of the conjugate gradient method

Replace α and β with the following form:

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k} \tag{3.14}$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \tag{3.15}$$

Rate of convergence

Among all possible methods whose first k steps are restricted to Krylov subspace, algorithm using 3.14 and 3.15 is the best one.

$$\frac{1}{2}||x - x^*||_A^2 = \frac{1}{2}(x - x^*)A(x - x^*) = \phi(x) - \phi(x^*) \tag{3.16}$$

We have

$$\begin{aligned} x_{k+1} - x^* &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \ldots + \gamma_k A^k r_0 - x^* \\ &= x_0 - x^* + (\gamma_0 I + \gamma_1 A + \ldots + \gamma_k A^k) A (x_0 - x^*) \\ &= [I + P_k(A)A](x_0 - x^*) \end{aligned} \tag{3.17}$$

where $P_k(A) = \gamma_0 I + \gamma_1 A + \ldots + \gamma_k A^k$.

Eigenvalue decomposition of A:

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T \tag{3.18}$$

Since the eigenvectors span the whole space R^n, we can write

$$x_0 - x^* = \sum_{i=1}^{n} \xi_i v_i \tag{3.19}$$

So 3.17 equals to

$$\sum_{i=1}^{n} \xi_i (1 + P_k(\lambda_i)\lambda_i) v_i \tag{3.20}$$

So we have

$$\begin{aligned} \phi(x_{k+1}) - \phi(x^*) &\le \max_{\lambda_1, \lambda_2, \ldots, \lambda_n} (1 + P_k(\lambda)\lambda)^2 \sum_{i=1}^{n} \xi_i^2 \lambda_i & (3.21) \\ &= \max_{\lambda_1, \lambda_2, \ldots, \lambda_n} (1 + P_k(\lambda)\lambda)^2 ||x_0 - x^*||_A^2 & (3.22) \end{aligned}$$

Theorem 3.4 *If A has only r distinct eigenvalues, then the CG iteration will terminate at the solution in at most r iterations.*

Theorem 3.5 *If A has eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$, we have*

$$||x_{k+1} - x^*||_A^2 \le (\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1})^2 ||x_0 - x^*||_A^2 \tag{3.23}$$

A is nice if its eigenvalues are clustered, or well-conditioned.

Preconditioning

Define new variable $\hat{x} = Cx$, then $\hat{\phi}(\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})\hat{x} - (C^{-T}b)^T\hat{x}$. Choose C so that $C^{-T}AC$ is well-conditioned or has clustered eigenvalues.

3.2 Nonlinear Conjugate Gradient methods

The Fletcher-Reeves method Extend the conjugate gradient method to nonlinear function by making two simple changes in 3.14. First, for the step length α_k, perform a line search that identifies an approximate minimum of the nonlinear function f along p_k. Second, the residual r is replaced by the gradient of the nonlinear objective f. That is, $\beta_{k+1}^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$

The Polak-Ribiere method and variants

Variants of FR method differ from each other mainly in the choice of parameter β. Polak-Ribiere defines this parameter as:

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{||f_k||^2} \tag{3.24}$$

Wolfe conditions do not guarantee that p_k is always a descent direction. If we define the β parameter as

$$\beta_{k+1}^{+} = \max\{\beta_{k+1}^{PR}, 0\} \tag{3.25}$$

In practice, PR+ performs more robust than FR, and their performances are influenced by the choice of the range of safeguarding.

Lemma 3.1 *Suppose that FR is implemented with a step length α_k that satisfies the strong Wolfe conditions with $0 < c_2 < \frac{1}{2}$. Then the method generates descent directions p_k that satisfy the following inequalities:*

$$-\frac{1}{1-c_2} \le \frac{\nabla f_k^T p_k}{||\nabla f_k||^2} \le \frac{2c_2 - 1}{1-c_2}, \textit{for all } k = 0, 1, \ldots \tag{3.26}$$

Theorem 3.6 *Suppose that f is bounded and Lipschitz continuously differentiable, and FR is implemented with a line search that satisfies the strong Wolfe conditions, with $0 < c_1 < c_2 < \frac{1}{2}$. Then*

$$\lim_{k\to\infty} \inf ||\nabla f_k|| = 0 \tag{3.27}$$

4 Quasi-Newton Methods

In Newton method, $p_k = -(\nabla^2 f(x))^{-1} \nabla f(x_k)$. In quasi Newton, use B_k to replace $\nabla^2 f(x_k)$ or H_k to replace $(\nabla^2 f(x_k))^{-1}$. That is, $p_k = -B_k^{-1} \nabla f(x)$ or $p_k = -H_k \nabla f(x_k)$.

Desired properties of B_k:

- don't use second derivations to compute it
- positive definite to guarantee descent
- symmetric
- behaves like true hessian
- B_{k+1} is small modification at B_k

Secant condition:

$$y_k = B_{k+1} s_k \tag{4.1}$$

Thus

$$s_k^T y_k > 0 \tag{4.2}$$

To determine B_{k+1} uniquely, we impose the additional condition that *among all symmetric matrices satisfying the secant equation, B_{k+1} is, in some sense, closest to the current matrix B_k.* That is,

$$\min_B ||B - B_k|| \text{ s.t. } B = B^T, B s_k = y_k \tag{4.3}$$

DFP updating formula:

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \tag{4.4}$$

with $\rho_k = \frac{1}{y_k^T s_k}$. The corresponding H is

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{4.5}$$

BFGS estimates H_k:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \tag{4.6}$$

It converges superlinearly.

4.1 The SR1 Method

In the BFGS and DFP, the updated matrix B_{k+1} differs from the predecessor B_k by a rand-2 matrix. SR1, *symmetric-rank-1* uses rank-1 update, but it does not guarantee that the updated matrix maintains positive definiteness. It has from

$$B_{k+1} = B_k + \sigma v v^T \tag{4.7}$$

As it satisfies the secant equation $y_k = B_k s_k$, we have

$$y_k = B_k + [\sigma v^T s_k] v \tag{4.8}$$

The term in brackets is a scalar, so v must be a multiple of $y_k - B_k s_k$. That is,

$$(y_k - B_k s_k) = \sigma \delta^2 [s_k^T (y_k - B_k s_k)](y_k - B_k s_k) \tag{4.9}$$

So the parameters should be

$$\sigma = \text{sign}[s_k^T (y_k - B_k s_k)], \delta = \pm |s_k^T (y_k - B_k s_k)|^{-1/2} \tag{4.10}$$

Hence, rank-1 updating formula is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \tag{4.11}$$

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \tag{4.12}$$

This is only defined when $(y_k - B_k s_k)^T s_k \neq 0$.

- Nice case when $(y_k - B_k s_k)^T s_k \neq 0$
- $y_k - B_k s_k = 0$, set $B_{k+1} = B_k$
- $(y_k - B_k s_k)^T s_k = 0$. skip the update

4.2 The Broyden class

The *Broyden Class* has a family of updates:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k (s_k^T B_k s_k) v_k v_k^T \tag{4.13}$$

where ϕ_k is a scalar parameter and

$$v_k = \left(\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}\right) \tag{4.14}$$

Each iteration, updates has form

$$p_k = -B_k^{-1} \nabla f_k, x_{k+1} = x_k + p_k \tag{4.15}$$

Set $\phi = 0$, we get BFGS; $\phi = 1$, we get DFP. $\phi = \frac{s_k^T y_k}{s_k^T (y_k - B_k s_k)}$, we get SR1.

Theorem 4.1 *Suppose that $f : R^n \to R$ is the strongly convex quadratic function $f(x) = b^T x + 1/2x^T Ax$, where A is symmetric and positive definite. Let x_0 be any starting point for the iteration B_k 4.15 and B_0 be any symmetric positive definite starting matrix, and suppose that the matrices B_k are updated by the Broyden formula 4.13 with $\phi_k \in [0, 1]$. Define $\lambda_1^k \leq \lambda_2^k \leq \ldots \lambda_n^k$ to be the eigenvalues of the matrix*

$$A^{1/2} B_k^{-1} A^{1/2} \tag{4.16}$$

Then for all k, we have

$$\min\{\lambda_i^k, 1\} \leq \lambda_i^{k+1} \leq \max\{\lambda_i^k, 1\} \tag{4.17}$$

Moreover, the property 4.17 does not hold if the Broyden parameter ϕ_k is chosen outside the interval [0,1].

5 Large-Scale Unconstrained Optimization

5.1 LBFGS

In BFGS, each step has the from

$$x_{k+1} = x_r - \alpha_k H_k \nabla f_k \tag{5.1}$$
$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \tag{5.2}$$
$$\rho_k = \frac{1}{y_k^T s_k}, V_k = I - \rho_k y_k s_k^T \tag{5.3}$$
$$s_k = x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k \tag{5.4}$$

Apply 5.2 recursively, and keep the last m steps:

$$\begin{aligned} H_k \quad = \quad & (V_{k-1}^T \dots V_{k-m}^T) H_k^0 (V_{k-m} \dots V_{k-1}) \\ & + \rho_{k-m} (V_{k-1}^T \dots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \dots V_{k-1}) \\ & + \rho_{k-m+1} (V_{k-1}^T \dots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \dots V_{k-1}) \\ & + \dots \\ & + \rho_{k-1} s_{k-1} s_{k-1}^T \end{aligned} \tag{5.5}$$

Algorithm 4 L-BFGS two-loop recursion

$q \leftarrow \nabla f_k$
for $i = k-1, k-2, \dots, k-m$ **do**
  $\alpha_i \leftarrow \rho_i s_i^T q$
  $q \leftarrow q - \alpha_i y_i$
end for
$r \leftarrow H_k^0 q$
for $i = k-m, k-m+1, \dots, k-1$ **do**
  $\beta \leftarrow \rho_i y_i^T r$
  $r \leftarrow r + s_i(\alpha_i - \beta)$
end for
stop with result $H_k \nabla f_k = r$

5.2 Inexact Newton Methods

Basic Newton step p_k^N is obtained by solving:

$$\nabla^2 f_k p_k^N = -\nabla f_k \tag{5.6}$$

The residual is

$$r_k = \nabla^2 f_k p_k + \nabla f_k \tag{5.7}$$

Algorithm 5 L-BFGS

Choose starting point x_0, integer $m > 0$
$k \leftarrow 0$
repeat
  Choose H_k^0
  Compute $p_k \leftarrow -H_k \nabla f_k$ from L-BFGS two-loop recursion
  Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k is chosen to satisfy the Wolfe conditions
  if $k > m$ **then**
    Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from storage
  end if
  Compute and save $s_k \leftarrow x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$
  $k \leftarrow k + 1$
until convergence

Use CG to solve this problem. Terminate the CG iterations when

$$||r_k|| \leq \eta_k ||\nabla f_k|| \tag{5.8}$$

Each iteration of CG requires us to compute $\nabla^2 f(x_k) v$ for some vector v. We approximate it by a finite difference

$$\nabla^2 f(x) v \approx 1/\epsilon [\nabla f(x_k + \epsilon v) - \nabla f(x_k)] \tag{5.9}$$

Then we perform a two levels of iteration-line search. In the outer loop, do line searches in directions given by the inner loop. In inner loop, use CG to calculate p_k.

Indefiniteness: CG only works if $\nabla^2 f(x_k)$ positive definite. We terminate the CG iteration as soon as a direction of negative curvature is generated.

Usually require $||r_k|| \leq \eta_k ||\nabla f(x_k)||$, where $0 \leq \eta_k \leq \eta < 1$.

Theorem 5.1 *Suppose that $\nabla^2 f(x)$ exists and is continuous in a neighborhood of a minimizer x^*, with $\nabla^2 f(x^*)$ is positive definite. Consider the iteration $x_{k+1} = x_k + p_k$, where p_k satisfies 5.8, and assume that $\eta_k \leq \eta$ for some constant $\eta \in [0, 1)$. Then, if the starting point x_0 is sufficiently near x^*, the sequence $\{x_k\}$ converges to x^* and satisfies*

$$||\nabla^2 f(x^*)(x_{k+1} - x^*)|| \leq \hat{\eta} ||\nabla^2 f(x^*)(x_k - x^*)|| \tag{5.10}$$

for some constant $\hat{\eta}$ with $\eta < \hat{\eta} < 1$.

In CG, it needs $\nabla^2 f_k d$, we can use the approximation

$$\nabla^2 f_k d \approx \frac{\nabla f(x_k + hd) - \nabla f(x_k)}{h} \tag{5.11}$$

We can use a trust-region framework for inexact Newton in place of line-search framework.

Outer loop: TR framework Inner loop: if $||r_{j+1}|| \leq \epsilon_k$ then stop inner iterations, set $p_k = z_{j+1}$.
Choose tolerance ϵ_k .
Generate CG steps $z_0, z_1, \ldots$
using CG search directions $d_0, d_1, d_2, \ldots$
Use $d_0 = -\nabla f(x_k)$
If $d_j^T \nabla^2 f(x_k) d_j \leq 0$, search along d_j: find a point that crosses TR boundary, use this as p_k.
Usually step $z_{j+1} = z_j + \alpha_j d_j$
but if $||z_{j+1}|| > \Delta$, then set step at TR boundary.
Set $p_i = z_{j+1}$

Theorem 5.2 *The sequence of vectors $\{z_j\}$ generated by algorithm above satisfies*

$$0 = ||z_0||_2 < \ldots < ||z_j||_2 < ||z_{j+1}||_2 < \ldots < ||p_k||_2 \leq \Delta_k \tag{5.12}$$

6 Derivative-Free Optimization

6.1 Finite Differences and Noise

Use finite-difference approximation to gradient:

$$\frac{\partial f}{\partial x_i} \approx \frac{f(x+\epsilon e_i) - f(x)}{\epsilon} \tag{6.1}$$

Through this, we can get a finite-difference approximation to gradient by doing n function evaluations. Can get a better gradient approximation using a centered difference formula.

$$\frac{\partial f}{\partial x_i} = \frac{f(x+\epsilon e_i) - f(x-\epsilon e_i)}{2\epsilon} + o(\epsilon^3) \tag{6.2}$$

In many applications, the objective function f has the form

$$f(x) = h(x) + \phi(x) \tag{6.3}$$

Define $\eta(x,\epsilon) = \max_{||z-x||_\infty \le \epsilon} |\phi(z)|$, then we have

$$|\frac{\partial h}{\partial x_i} - \frac{f(x+\epsilon e_i) - f(x)}{\epsilon}| = o(\epsilon) + \frac{2\eta(x;\epsilon)}{\epsilon} \tag{6.4}$$

Thus error bounded by $M\epsilon + \frac{\eta}{2}$. Choose ϵ to minimize this min:

$$\epsilon = \sqrt{\frac{\eta}{M}} \tag{6.5}$$

6.2 Model-Based methods

Wish to construct a quadratic model of the form

$$m_k(x_k + p) = c + g^T p + 1/2 p^T G p \tag{6.6}$$

We want $m_k(x_k+p) = f(x_k+p)$ at a bunch of p values:

$$m_k(y^l) = f(y^l), l = 1, 2, \ldots, q \tag{6.7}$$

As there are $1/2(n+1)(n+2)$ coefficients (c, g,and$G$ taking into account the symmetry of G), the interpolation conditions determine m_k uniquely only if $q = 1/2(n+1)(n+2)$.

An alternative is that just do $n+1$ initial evaluations and construct linear model with $G = 0$. Take steps based on the linear model, after accumulating f values at enough points, switch to quadratic model.

Replacing one member of $\{y^1, y^2, \ldots, y^q\}$ equals to replacing one row of matrix Y, and we would like to do this in a way that makes Y more nonsingular. Use determinant $\delta(\eta)$ to measure nonsingularty. In doing replacement, try to increase $|\delta(\eta)|$.

Perform a trust region. In each step, if the improvement is less than η, shrink Δ_k, improve $\{y^1, y^2, \ldots, y^q\}$

Idea is to do a least-change modification to G:

$$\begin{aligned} \min_{f,g,G} &||G - G_k||_F^2 \\ \text{s.t. } &G \text{ symmetric} \\ m(y^l) = f(y^l), &l = 1, 2, \ldots, \hat{q} \end{aligned} \tag{6.8}$$

6.3 Coordinate Search and Pattern-Search Methods

Coordinate Search Search along different coordinate directions $x \in \mathcal{R}^n$. Search directions e_i, need to cycle though all $i = 1, \ldots, n$

Pattern Search At each x_k, we have a set of possible directions $\mathcal{D}_k$, search along some or all directions in $\mathcal{D}_k$ until we find a point $x_k + \gamma_k p_k$ for $\gamma > 0$ with "sufficiently better" f value $p_k \in \mathcal{D}_k$. If no $p_k \in \mathcal{D}_k$ have $f(x_k + \gamma_k p_k) = f(x_k) - \rho(\gamma_k)$, decrease γ_k and repeat.

Validity requirement: for any $v \in \mathcal{R}^n$ that there is at least one $p \in \mathcal{D}_k$ such that $p^T v > 0$.

$$\kappa(\mathcal{D}_k) = \min_{v \in R^n} \max_{p \in \mathcal{D}_k} \frac{v^T p}{||v||||p||} \geq \delta \tag{6.9}$$

References

[1] Stephen J. Wright Jorge Nocedal. *Numerical Optimization*. Springer, 2 edition, 2006.