第16讲 大规模无约束优化

- 非精确Newton-CG方法
- 信赖域Newton-CG方法
- 有限记忆BFGS方法

大规模无约束优化问题： $\min\limits_{x\in R^n} f(x)$

古典Newton法的迭代公式: $x_{k+1} = x_k + d_k$

其中x_k处的Newton方向

$$d_k = -\nabla^2 f(x_k)^{-1}\nabla f(x_k)$$

等价于求解方程组

$$\nabla^2 f(x_k)d + \nabla f(x_k) = 0$$

如果方程组没有精确计算到解，牛顿方法是否还有类似的收敛性和收敛速度？

局部收敛的非精确牛顿法

迭代格式：$x_{k+1} = x_k + p_k$

其中 p_k满足： $\|\nabla^2 f(x_k)p + \nabla f(x_k)\| \le \eta_k \|\nabla f(x_k)\|$,　　（7.3）

且 $0 < \eta_k < 1$

Theorem 7.1.

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 (7.3), and assume that $\eta_k \le \eta$ 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

$$\|\nabla^2 f(x^*)(x_{k+1} - x^*)\| \le \hat{\eta}\|\nabla^2 f(x^*)(x_k - x^*)\|, \tag{7.4}$$

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

Theorem 7.2.

Suppose that the conditions of Theorem 7.1 hold, and assume that the iterates $\{x_k\}$ generated by the inexact Newton method converge to x^*. Then the rate of convergence is superlinear if $\eta_k \to 0$. If in addition, $\nabla^2 f(x)$ is Lipschitz continuous for x near x^* and if $\eta_k = O(\|\nabla f_k\|)$, then the convergence is quadratic.

To obtain superlinear convergence, we can set, for example, $\eta_k = \min\left(0.5, \sqrt{\|\nabla f_k\|}\right)$; the choice $\eta_k = \min(0.5, \|\nabla f_k\|)$ would yield quadratic convergence.

Algorithm 7.1 (Line Search Newton–CG).

Given initial point x_0;
for $k = 0, 1, 2, \ldots$
  Define tolerance $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|})\|\nabla f_k\|$;
  Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
  for $j = 0, 1, 2, \ldots$
    if $d_j^T B_k d_j \leq 0$
      if $j = 0$
        return $p_k = -\nabla f_k$;
      else
        return $p_k = z_j$;
    Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
    Set $z_{j+1} = z_j + \alpha_j d_j$;
    Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
    if $\|r_{j+1}\| < \epsilon_k$
      return $p_k = z_{j+1}$;
    Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
    Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
  end (for)
  Set $x_{k+1} = x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or
    Armijo backtracking conditions (using $\alpha_k = 1$ if possible);
end

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

信赖域Newton-CG方法

Algorithm 4.1 (Trust Region).

Given $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in \left[0, \frac{1}{4}\right)$:

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

Obtain p_k by (approximately) solving (4.3);

Evaluate ρ_k from (4.4);

if $\rho_k < \frac{1}{4}$

$\Delta_{k+1} = \frac{1}{4}\Delta_k$

else

if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$

$\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$

else

$\Delta_{k+1} = \Delta_k$;

if $\rho_k > \eta$

$x_{k+1} = x_k + p_k$

else

$x_{k+1} = x_k$;

end (for).

信赖域子问题的求解

Algorithm 7.2 (CG–Steihaug).

Given tolerance $\epsilon_k > 0$;
Set $z_0 = 0, r_0 = \nabla f_k, d_0 = -r_0 = -\nabla f_k$;
if $\|r_0\| < \epsilon_k$
  return $p_k = z_0 = 0$;
for $j = 0, 1, 2, \ldots$
  if $d_j^T B_k d_j \le 0$
    Find τ such that $p_k = z_j + \tau d_j$ minimizes $m_k(p_k)$ in (4.5)
      and satisfies $\|p_k\| = \Delta_k$;
    return p_k;
  Set $\alpha_j = r_j^T r_j / d_j^T B_k d_j$;
  Set $z_{j+1} = z_j + \alpha_j d_j$;
  if $\|z_{j+1}\| \ge \Delta_k$
    Find $\tau \ge 0$ such that $p_k = z_j + \tau d_j$ satisfies $\|p_k\| = \Delta_k$;
    return p_k;
  Set $r_{j+1} = r_j + \alpha_j B_k d_j$;
  if $\|r_{j+1}\| < \epsilon_k$
    return $p_k = z_{j+1}$;
  Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
  Set $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$;
end (for).

有限记忆BFGDS

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k,$$

Algorithm 7.5 (L-BFGS).

Choose starting point x_0, integer $m > 0$;

$k \leftarrow 0$;

repeat

Choose H_k^0 (for example, by using (7.20));

Compute $p_k \leftarrow -H_k \nabla f_k$ from Algorithm 7.4;

Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k is chosen to satisfy the Wolfe conditions;

if $k > m$

Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from storage;

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.

有限记忆BFGDS

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T$$

$$\text{(BFGS)} \qquad H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \qquad (6.17)$$

$$\rho_k = \frac{1}{y_k^T s_k}, \qquad V_k = I - \rho_k y_k s_k^T, \qquad s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k.$$

$$\boxed{\begin{aligned} & H^k \\ = & (V^{k-m} \cdots V^{k-1})^{\mathrm{T}} H^{k-m} (V^{k-m} \cdots V^{k-1}) + \\ & \rho_{k-m} (V^{k-m+1} \cdots V^{k-1})^{\mathrm{T}} s^{k-m} (s^{k-m})^{\mathrm{T}} (V^{k-m+1} \cdots V^{k-1}) + \\ & \rho_{k-m+1} (V^{k-m+2} \cdots V^{k-1})^{\mathrm{T}} s^{k-m+1} (s^{k-m+1})^{\mathrm{T}} (V^{k-m+2} \cdots V^{k-1}) + \\ & \cdots + \rho_{k-1} s^{k-1} (s^{k-1})^{\mathrm{T}}. \end{aligned}}$$

$$
\begin{aligned}
& H^k \\
=& (V^{k-m}\cdots V^{k-1})^{\mathrm{T}} H^{k-m} (V^{k-m}\cdots V^{k-1}) + \\
& \rho_{k-m} (V^{k-m+1}\cdots V^{k-1})^{\mathrm{T}} s^{k-m} (s^{k-m})^{\mathrm{T}} (V^{k-m+1}\cdots V^{k-1}) + \\
& \rho_{k-m+1} (V^{k-m+2}\cdots V^{k-1})^{\mathrm{T}} s^{k-m+1} (s^{k-m+1})^{\mathrm{T}} (V^{k-m+2}\cdots V^{k-1}) + \\
& \cdots + \rho_{k-1} s^{k-1} (s^{k-1})^{\mathrm{T}}.
\end{aligned}
$$

左右两边同时右乘 $\nabla f(x_k)$

$$
V^{k-1}\nabla f(x^k),\ V^{k-2}V^{k-1}\nabla f(x^k),\ \cdots,\ V^{k-m}\cdots V^{k-2}V^{k-1}\nabla f(x^k).
$$

$$
\begin{aligned}
H^k\nabla f(x^k) =& (V^{k-m}\cdots V^{k-1})^{\mathrm{T}} H^{k-m} q + \\
& (V^{k-m+1}\cdots V^{k-1})^{\mathrm{T}} s^{k-m} \alpha_{k-m} + \\
& (V^{k-m+2}\cdots V^{k-1})^{\mathrm{T}} s^{k-m+1} \alpha_{k-m+1} + \cdots + s^{k-1}\alpha_{k-1}.
\end{aligned}
$$

$$
V^k = I - \rho_k y^k (s^k)^{\mathrm{T}}
$$

Algorithm 7.4 (L-BFGS two-loop recursion).

$q \leftarrow \nabla f_k$;

for $i = k-1, k-2, \ldots, k-m$

$\quad \alpha_i \leftarrow \rho_i s_i^T q$;

$\quad q \leftarrow q - \alpha_i y_i$;

end (for)

$r \leftarrow H_k^0 q$;

for $i = k-m, k-m+1, \ldots, k-1$

$\quad \beta \leftarrow \rho_i y_i^T r$;

$\quad r \leftarrow r + s_i(\alpha_i - \beta)$

end (for)

stop with result $H_k \nabla f_k = r$.

$$\boxed{\begin{aligned} H^k \nabla f(x^k) = &(V^{k-m} \cdots V^{k-1})^{\mathrm{T}} H^{k-m} q + \\ &(V^{k-m+1} \cdots V^{k-1})^{\mathrm{T}} s^{k-m} \alpha_{k-m} + \\ &(V^{k-m+2} \cdots V^{k-1})^{\mathrm{T}} s^{k-m+1} \alpha_{k-m+1} + \cdots + s^{k-1} \alpha_{k-1}. \end{aligned}}$$

$$\begin{aligned} &(V^{k-m+1} \cdots V^{k-1})^{\mathrm{T}}((V^{k-m})^{\mathrm{T}} r + \alpha_{k-m} s^{k-m}) \\ = &(V^{k-m+1} \cdots V^{k-1})^{\mathrm{T}}(r + (\alpha_{k-m} - \beta) s^{k-m}), \end{aligned}$$

$$\boxed{V^k = I - \rho_k y^k (s^k)^{\mathrm{T}}}$$

近似矩阵的取法

$$H_k^0 = \gamma_k I,$$

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}.$$

练习: 习题6

4. 用信赖域算法 6.1 和子问题的截断共轭梯度算法 6.4 编程计算如下最优化问题 (取 $n = 10$) 的解.

$$\min \ f(x) = \sum_{i=1}^{n} \left[(1 - x_{2i-1})^2 + 10(x_{2i} - x_{2i-1}^2)^2 \right].$$

实现非精确牛顿法Algorithm 7.1 和信赖域Newton-CG法（Algorithm7.2 ），L-BFGS（Algorithm7.5），
对上面的问题计算n=5000,10000,100000, 进行比较。

微分计算

- 有限差分
- 自动差分
- 符号差分

有限差分导数逼近

- 主要工具: Talyer 展式

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p, \qquad \text{some } t \in (0,1)$$

- 向前差分：

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

有限差分导数逼近

- 取 $p = \varepsilon e_i$

$$\nabla f(x)^T p = \nabla f(x)^T e_i = \partial f / \partial x_i,$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x)}{\epsilon} + \delta_\epsilon, \qquad \text{where } |\delta_\epsilon| \le (L/2)\epsilon.$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} + O\left(\epsilon^2\right).$$

By setting $p = \epsilon e_i$ and $p = -\epsilon e_i$, respectively, we obtain

$$f(x + \epsilon e_i) = f(x) + \epsilon \frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2 \frac{\partial^2 f}{\partial x_i^2} + O\left(\epsilon^3\right),$$

$$f(x - \epsilon e_i) = f(x) - \epsilon \frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2 \frac{\partial^2 f}{\partial x_i^2} + O\left(\epsilon^3\right).$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} + O\left(\epsilon^2\right).$$

$\mathbf{u}$ known as *unit roundoff*

$$|\text{comp}(f(x)) - f(x)| \le \mathbf{u}L_f,$$
$$|\text{comp}(f(x+\epsilon e_i)) - f(x+\epsilon e_i)| \le \mathbf{u}L_f,$$

$$\boxed{\frac{\partial f}{\partial x_i}(x) = \frac{f(x+\epsilon e_i) - f(x)}{\epsilon} + \delta_\epsilon, \qquad \text{where } |\delta_\epsilon| \le (L/2)\epsilon. \qquad (8.4)}$$

an error that is bounded by

$$(L/2)\epsilon + 2\mathbf{u}L_f/\epsilon.$$

the minimizing value is

$$\epsilon^2 = \frac{4L_f\mathbf{u}}{L}.$$

$$\boxed{\epsilon = \sqrt{\mathbf{u}}.}$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} + O\left(\epsilon^2\right).$$

最优取值 $\varepsilon = u^{\frac{1}{3}}$

误差界 $\mathrm{u}^{\frac{2}{3}}$

$$\nabla f(x + \epsilon p) = \nabla f(x) + \epsilon \nabla^2 f(x) p + O(\epsilon^2), \tag{8.19}$$

so that

$$\nabla^2 f(x) p \approx \frac{\nabla f(x + \epsilon p) - \nabla f(x)}{\epsilon} \tag{8.20}$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{f(x + \epsilon e_i + \epsilon e_j) - f(x + \epsilon e_i) - f(x + \epsilon e_j) + f(x)}{\epsilon^2} + O(\epsilon). \tag{8.21}$$

复合优化问题的算法

$$\min_{x\in\mathbb{R}^n} \quad \psi(x) = f(x) + h(x),$$

主要方法：

（具体见《最优化：建模、算法与理论》第八章）

近似点梯度法

Nesterov加速算法

近似点算法

分块坐标下降法

随机优化算法