第二讲: Line Search Methods (线搜索方法)

GENERAL DESCRIPTION

- 一般迭代格式为$x_{k+1} = x_k + \alpha_k p_k$ 关键是构造搜索方向p_k和步长因子α_k.
- 设$\varphi(\alpha) = f(x_k + \alpha p_k)$, 沿着$p_k$, 确定步长因子$\alpha_k$使得 $\varphi(\alpha_k) < \varphi(0)$.
  - $\alpha_k = \arg\min_{\alpha>0} \varphi(\alpha)$ 称为最优线搜索或精确线搜索，或最优一维搜索.
  - 如果α_k, 使目标函数f得到可接受的下降量, 即使得下降量 $f(x_k) - f(x_k + \alpha_k p_k) > 0$ 是可以接受的, 则称这样的一维搜索为近似一维搜索，或不精确一维搜索.
- 一维搜索主要结构:
  - 首先确定包含问题最优解得搜索区间,
  - 采用某种分割技术或插值方法缩小这个区间, 进行搜索.
- 设α^* 是满足 $\varphi(\alpha^*) = \min_{\alpha\geq 0} \varphi(\alpha)$. 如果存在$[a, b] \subset [0, \infty)$, 使得$\alpha^* \in [a, b]$, 则称$[a, b]$是一维极小化$\min_{\alpha\geq 0} \varphi(\alpha)$的搜索区间.
- 确定搜索区间的一种简单方法：进退法。基本思想是从一点出发，按一定步长，试图确定出函数值呈现"高-低-高"三点. 一个方向不成功, 就退回来, 再沿相反方向寻找.

GENERAL DESCRIPTION

进退法搜索

1. 选取初始数据. 给定α_0, $h_0 > 0$, 加倍系数$t > 1$, 计算$\varphi(\alpha_0)$, 设$k = 0$;
2. 比较目标函数值. 令$\alpha_{k+1} = \alpha_k + h_k$, 计算$\varphi_{k+1} = \varphi(\alpha_{k+1})$,
如果$\varphi_{k+1} < \varphi_k$, 转步3, 否则转步4
3. 加大搜索步长. 令$h_{k+1} = th_k$, $\alpha = \alpha_k$, $\alpha_k = \alpha_{k+1}$, $\varphi_k = \varphi_{k+1}$, $k = k + 1$,
转步2.
4. 反向探索. 若$k = 0$, 转换探索方向, 令$h_k := -h_k$, $\alpha_k = \alpha_{k+1}$, 转步2;
否则, 停止迭代, 令

$$a = \min\{\alpha, \alpha_{k+1}\}, \quad b = \max\{\alpha, \alpha_{k+1}\}.$$

定义单峰/谷函数(unimodal function)

设$\varphi : R \to R$, $[a, b] \subset R$, 若存在$\alpha^* \in [a, b]$, 使得$\varphi(\alpha)$ 在$[a, \alpha^*]$上严格递减,
在$[\alpha^*, b]$上严格递增, 则称$[a, b]$是函数φ的单峰区间(或单谷区间).

精确一维搜索

算法2.1

给定$x_0 \in R^n$, $0 \le \varepsilon \ll 1$;

for $k = 0, 1, \cdots$

  计算搜索方向p_k;

  计算步长α_k, 使得 $f(x_k + \alpha_k p_k) = \min_{\alpha \ge 0} f(x_k + \alpha p_k)$;

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

  if $\|\nabla f(x_k)\| \le \varepsilon$

    stop;

  end (if)

end (for)

定义向量之间的夹角

设$\theta_k = \langle p_k, \nabla f(x_k)\rangle$表示向量$p_k$和向量$\nabla f(x_k)$之间的夹角，则有

$$\cos\theta_k = \cos\langle p_k, \nabla f(x_k)\rangle = \frac{p_k^T \nabla f(x_k)}{\|p_k\|\|\nabla f(x_k)\|}.$$

0.618法、FIBONACCI法和二分法

- 基本思想： 通过取试探点进行函数值比较，使得包含极小值点的搜索区间不断缩短, 当区间长度缩短到一定程度时, 区间上个点均接近极小值. 仅需计算函数值，不需要计算导数值， 适用于非光滑及导数表达式复杂的或写不出的情形。
- 设$\varphi(\alpha) = f(x_k + \alpha p_k)$，是搜索区间$[a_1, b_1]$上的单峰函数.
- 假设在k次迭代时搜索区间为$[a_k, b_k]$. 取两个试探点$\lambda_k, \mu_k \in [a_k, b_k]$, 且$\lambda_k < \mu_k$,要求满足下列条件：
  1. λ_k和μ_k到搜索区间$[a_k, b_k]$两端点等距，即 $b_k - \lambda_k = \mu_k - a_k$.
  2. 每次迭代，搜索区间长度缩短率相同, 即$b_{k+1} - a_{k+1} = \tau(b_k - a_k)$.
- 如果$\varphi(\lambda_k) \le \varphi(\mu_k)$, 则令$a_{k+1} = a_k$, $b_{k+1} = \mu_k$.
  如果$\varphi(\lambda_k) > \varphi(\mu_k)$, 则令$a_{k+1} = \lambda_k$, $b_{k+1} = b_k$.
- $\tau = \frac{\sqrt{5}-1}{2} \approx 0.618$. (黄金分割法)
  $\lambda_k = a_k + 0.382(b_k - a_k)$, $\quad \mu_k = a_k + 0.618(b_k - a_k)$.

0.618法、FIBONACCI法和二分法

- Fibonacci法中τ不在是常数而是$\tau_k = \frac{F_{n-k}}{F_{n-k+1}}$, 其中
- Fibonacci数列 $F_0 = F_1 = 1, \quad F_{k+1} = F_k + F_{k-1}, k = 1, 2 \cdots,$.
- $\lambda_k = a_k + (1 - \tau_k)(b_k - a_k) = a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k)$
  $\mu_k = a_k + \tau_k(b_k - a_k) = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)$
- 假设$F_k \approx r^k$, 有 $r^{k+1} = r^k + r^{k-1}$ 可以推出 $r = \frac{\sqrt{5}-1}{2}$.即 Fibonacci法渐进行为就是黄金分割法.
- 事实上, 可以证明Fibonacci法是分割方法求解一维极小化问题的最优策略, 而黄金分割法是近似最优法.
- 二分法$\lambda_k = \mu_k = \frac{a_k+b_k}{2}$.
- 分割法都是线性收敛的方法。

插值法

- 基本思想：在搜索区间中不断使用低次多项式来近似目标函数，并逐步用插值多项式的极小点来逼近一维搜索问题$\min\limits_{\alpha} \varphi(\alpha)$的极小点.
- 当函数解析性质比较好时，插值法比分割法效果更好.
- 二次插值法（单点，二点，三点），局部二阶收敛、超线性收敛
- 三次插值法（二点），局部二阶收敛

单点插值法(牛顿法)

- 考虑利用某一点处的函数值、一阶导数值、二阶导数值构造二次函数
- 设$q(\alpha) = a\alpha^2 + b\alpha + c$
  满足 $q(\alpha_1) = \varphi(\alpha_1)$, $q'(\alpha_1) = \varphi'(\alpha_1)$, $q''(\alpha_1) = \varphi''(\alpha_1)$.
- 直接求解$q(\alpha)$的最小值可得：$\bar{\alpha} = -\frac{b}{2a} = \alpha_1 - \frac{\varphi'(\alpha_1)}{\varphi''(\alpha_1)}$.
- 本质上是牛顿法。（具有局部的二次收敛性）

不精确一维搜索法

- 一维搜索是最优化方法的基本组成部分
- 精确的一维搜索花费巨大
- 很多最优化方法，例如牛顿法/拟牛顿法，收敛速度不依赖于精确一维搜索过程

不精确一维搜索法

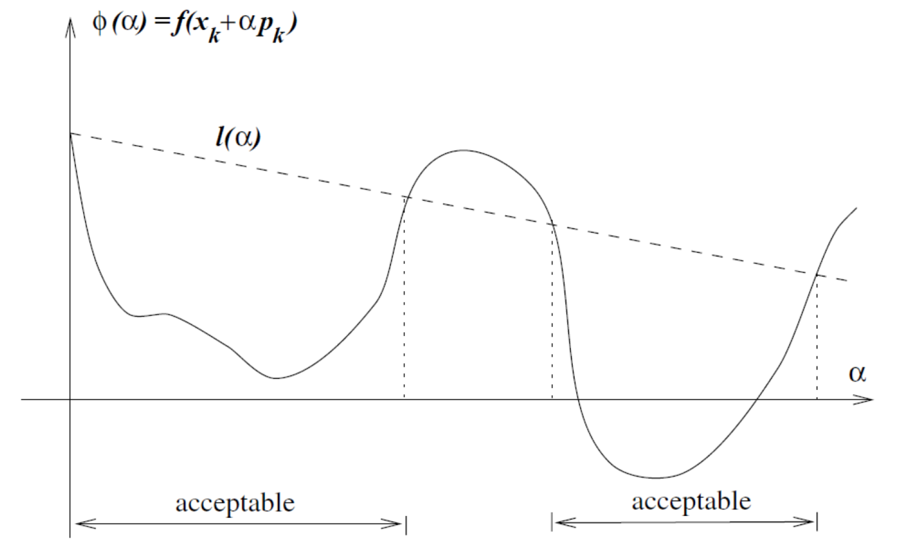
Armijo condition: 首先保证 α_k 能够使目标函数f产生足够下降 sufficient decrease

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f^T(x_k) p_k \tag{2.1}$$

for some constant $c_1 \in (0, 1)$. In practice, c_1 is chosen to be quite small, say $c_1 = 10^{-4}$.

(2.1) means that the reduction in f should be proportional to both the step length α_k and the directional derivative $\nabla f^T(x_k) p_k$.

Demo: Sufficient Decrease Condition



The Wolfe 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 α.
- To rule out unacceptably short steps we introduce a second requirement, called the *curvature condition*, which requires α_k to satisfy

$$\left(\nabla f(x_k + \alpha_k p_k)\right)^T p_k \geq c_2(\nabla f(x_k))^T p_k \tag{2.2}$$

  for some constant $c_2 \in (c_1, 1)$, where c_1 (通常很小) is the constant from (2.1), i.e.,

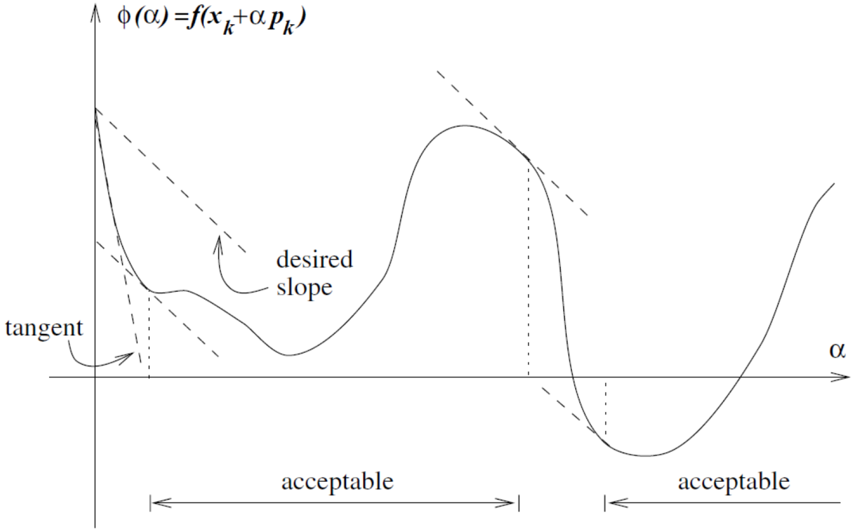
$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla^T(x_k) p_k$$

- Typical values of $c_2 \approx 0.9$ when the search direction p_k is chosen by a Newton or quasi-Newton method, or $c_2 \approx 0.1$ when p_k is obtained from a nonlinear conjugate gradient method.

THE WOLFE CONDITION

- Note that the left-hand-side is simply the derivative $\phi'(\alpha_k)$, so the curvature condition ensures that the slope of ϕ at α_k is greater than c_2 times the initial step slope $\phi'(0)$, i.e., $\phi'(\alpha_k) \geq c_2\phi'(0)$.
- 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.
- On the other hand, 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.

The Wolfe Condition



The Wolfe Condition

The sufficient decrease and the curvature conditions are known collectively as the Wolfe conditions. We restate them here for future reference:

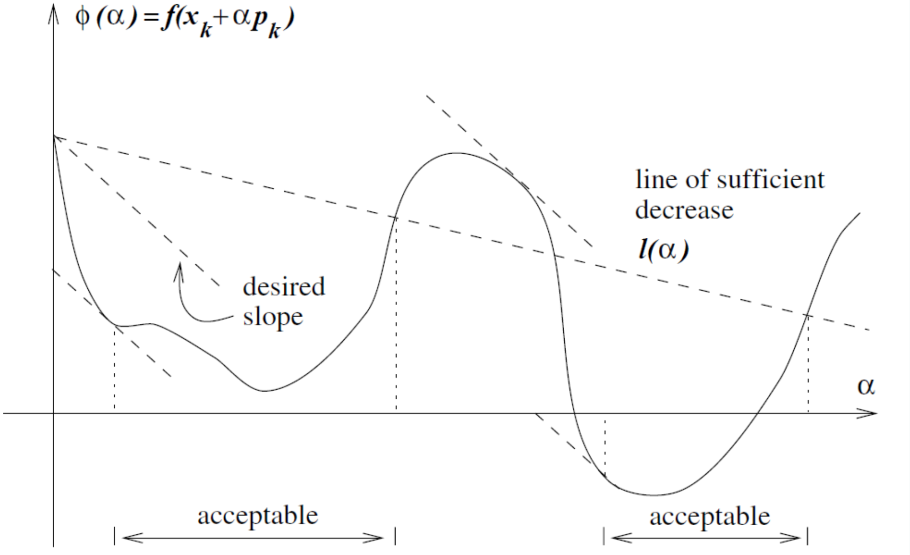
$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k (\nabla f(x_k))^T p_k \tag{2.3a}$$

$$(\nabla f(x_k + \alpha_k p_k))^T p_k \geq c_2 (\nabla f(x_k))^T p_k \tag{2.3b}$$

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 Condition



STRONG WOLFE CONDITION

- 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

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k (\nabla f(x_k))^T p_k \tag{2.4a}$$

$$|(\nabla f(x_k + \alpha_k p_k))^T p_k| \le c_2 |(\nabla f(x_k))^T p_k| \tag{2.4b}$$

  with $0 < c_1 < c_2 < 1$.
- The only difference with the Wolfe condition is that we no longer allow the derivative $\phi'(\alpha_k)$ to be too positive. Hence, we exclude points that are far from stationary points of ϕ.

The Goldstein Condition

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

$$f(x_k) + (1-c)\alpha_k(\nabla f(x_k))^T p_k \le f(x_k + \alpha_k p_k) \le f(x_k) + c\alpha_k(\nabla f(x_k))^T p_k, \tag{2.5}$$

with $0 < c < \frac{1}{2}$.

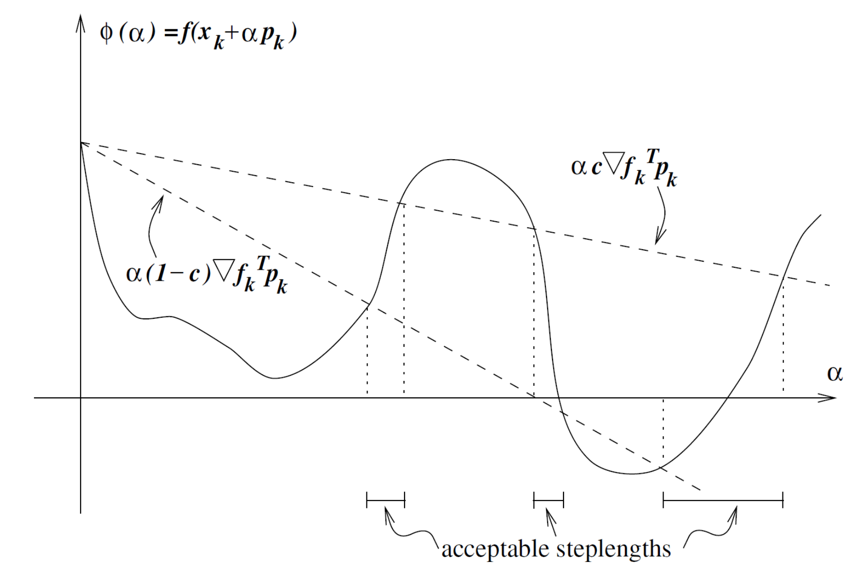
- The second equality is the sufficient decrease condition (2.1)
- The first inequality is introduced to control the step length from below.

The Goldstein Condition

排除

- A disadvantage of the Goldstein conditions vs the Wolfe conditions is that the first inequality in (2.5) 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 no well suited for quasi-Newton methods, which maintain a positive definite Hessian approximation.

THE GOLDSTEIN CONDITION



STEP-LENGTH SELECTION ALGORITHMS

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

$$\alpha_k = -\frac{(\nabla f(x_k))^T p_k}{p_k Q p_k}$$

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

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.

Initial Step Length

- A popular strategy is to assume that the first-order change in the function at iterate x_k will be the same as that obtained at the previous step.
  In other words, we choose the initial guess α_0, so that $\alpha_0 \nabla f(x_k)^T p_k = \alpha_{k-1} \nabla f(x_{k-1})^T p_{k-1}$, that is,

$$\alpha_0 = \alpha_{k-1} \frac{\nabla f(x_{k-1})^T p_{k-1}}{\nabla f(x_k)^T p_k} \tag{2.9}$$

INITIAL STEP LENGTH

- Another useful strategy: interpolate a quadratic to the data $f(x_{k-1}), f(x_k)$, and $\phi'(0) = \nabla f(x_{k-1})^T p_{k-1}$ and define α_0 to be its minimizer.
- This strategy yields

$$\alpha_0 = \frac{2(f(x_k) - f(x_{k-1}))}{\phi'(0)} \tag{2.10}$$

- It can be shown that if $x_k \to x^*$ superlinearly, then the ratio in this expression converges to 1. If we adjust the choice (2.10) by setting

$$\alpha_0 \leftarrow \min(1, 1.01\alpha_0)$$

  we find that the unit step length $\alpha_0 = 1$ will eventually always be tried and accepted, and the superlinear convergence properties of Newton and quasi-Newton methods will be observed.

Convergence of Line Search Methods

- We discuss requirements on the search direction in this section.
- Focusing on one key property: the angle between p_k and the steepest descent direction $-\nabla f(x_k)$, defined by θ_k

$$\cos\theta_k = \frac{-\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|\|p_k\|} \tag{2.11}$$

Convergence of Line Search Methods

Theorem (Zoutendijk)

- Consider any iteration of the form (2.19), where p_k is a descent direction and α_k satisfies the Wolfe conditions (2.3).
- Suppose that $f(x)$ is bounded below in $\mathcal{R}^n$ and that $f(x)$ is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{N} \equiv \{x| : f(x) \leq f(x_0)\}$, 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

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{N}. \tag{2.12}$$

- **Then**

$$\sum_{k\geq 0} \cos^2(\theta_k)\|\nabla f(x_k)\|^2 < \infty \tag{2.13}$$

which is called Zoutendijk condition.

Convergence of Line Search Methods

Remark

- 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 (2.13) implies that

$$\cos^2(\theta_k)\|\nabla f(x_k)\|^2 \to 0. \tag{2.14}$$

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

CONVERGENCE OF LINE SEARCH METHODS

REMARK

- If the search direction p_k is chosen that the angle θ_k is bounded away from 90°, there is a positive constant δ such that

$$\cos\theta_k \geq \delta > 0, \forall k \tag{2.15}$$

It follows immediately from (2.14) that

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0. \tag{2.16}$$

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