法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

无约束优化问题

无约束优化问题（unconstrained optimization）

f'

$$\min_{x\in\mathbb{R}^n} f(x)$$

$x^*.\quad f(x^*).$

$f:\mathbb{R}^n\to\mathbb{R}$ 光滑函数（smooth function）

- 称x^*为全局最优点，如果$f(x^*)\le f(x),\forall x\in\mathbb{R}^n$

最小值

- 称x^*为局部最优点，如果存在x^*的一个邻域$\mathcal{N}$使得$f(x^*)\le f(x),\forall x\in\mathcal{N}$

极小值

无约束优化问题

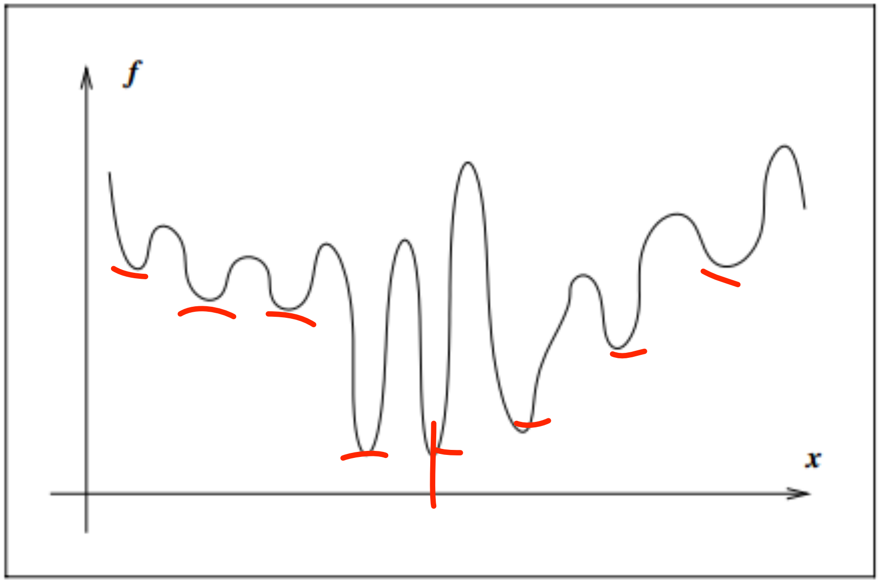


Figure 2.2 A difficult case for global minimization.

遗传算法

粒子群算法

Figure 2.2 illustrates a function with many local minimizers. It is usually difficult to find the global minimizer for such functions, because algorithms tend to be “trapped” at local minimizers. This example is by no means pathological. In optimization problems associated with the determination of molecular conformation, the potential function to be minimized may have millions of local minima.

无约束优化问题

f'

Theorem 2.2 (First-Order Necessary Conditions).

If x^* is a local minimizer and f is continuously differentiable in an open neighborhood of x^*, then $\nabla f(x^*) = 0$.

Theorem 2.3 (Second-Order Necessary Conditions).

If x^* is a local minimizer of f and $\nabla^2 f$ exists and is continuous in an open neighborhood of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

Theorem 2.4 (Second-Order Sufficient Conditions).

Suppose that $\nabla^2 f$ is continuous in an open neighborhood of x^* and that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then x^* is a strict local minimizer of f.

Theorem 2.5.

f

When f is convex, any local minimizer x^* is a global minimizer of f. If in addition f is differentiable, then any stationary point x^* is a global minimizer of f.

$\nabla f = 0$

无约束优化问题

线搜索方法（line search）

构造一系列收敛序列$\{x_k\}$收敛到最优点x^*，每次对于当前的状态x_k，选择优化方向p_k，以及优化步长α，最终使得

$x_k \to x^*$

$$\min_{\alpha>0} f(x_k + \alpha p_k)$$

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

置信域方法（trust region）

在x_k附近构造一个模型函数（model function）m_k作为f在x_k附近的一个估计，当然这个估计只能在局部是比较好的，然后我们去求解下述子问题

$$\min_{p} m_k(x_k + p), \qquad where\ x_k + p\ lies\ in\ the\ trust\ region$$

线搜索

f　p_k

在线搜索(line search)方法中，主要的迭代为

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

成功的线搜索方法取决于方向p_k和步长(step length)α_k的选择。

大多数线搜索算法中，要求 p_k 是一个下降方向(descent direction)，即$p_k^T \nabla f_k < 0$，一般而言，要求线搜索的方向有下述形式

∇f_k　$\nabla f(x)|_{x=x_k}$

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

$\langle a, b\rangle$

$|a||b|\cos\langle a, b\rangle$

其中B_k是一个对称的满秩矩阵（symmetric and nonsingular）

- steepest descent method: I
- Newton's method: the exact Hessian $\nabla^2 f(x_k)$
- Quasi-Newton method: an approximation to the Hessian that is updated at every iteration by means of a low-rank formula.

线搜索

LINESEARCH METHODS

- calculate a **search direction** p_k from x_k
- ensure that this direction is a **descent direction**, i.e.,

$$g_k^T p_k < 0 \text{ if } g_k \neq 0$$

so that, for small steps along p_k, the objective function **will** be reduced

- calculate a suitable **steplength** $\alpha_k > 0$ so that

$$f(x_k) = \quad f(x_k + \alpha_k p_k) < f_k$$

- computation of α_k is the **linesearch**—may itself be an iteration
- generic linesearch method:

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

步长

在确定了搜索方向p_k的情况下，我们来讨论每次的步长α_k，理想的情况我们希望对下述函数求到最小值

$$\phi(\alpha) = f(x_k + \alpha p_k)$$

直接求解上述函数的最小值的方法叫做精确线搜索，然而这需要耗费大量的计算，所以我们一般采用非精确线搜索(inexact line search)

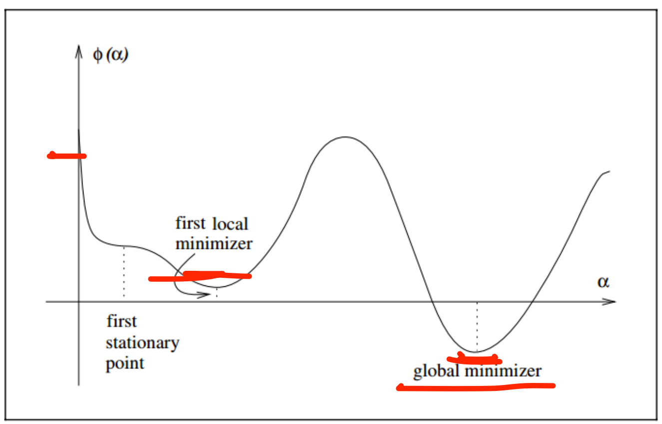
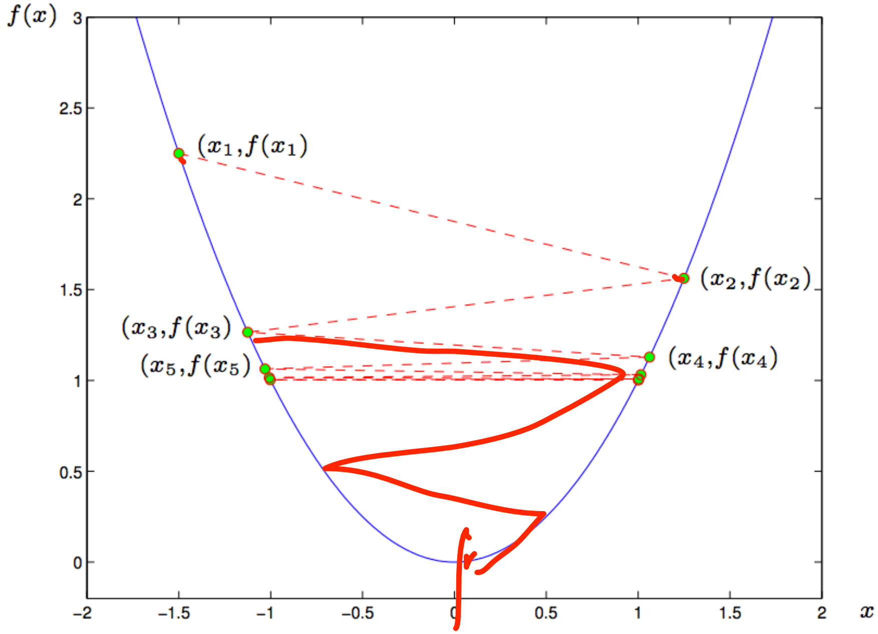


Figure 3.1 The ideal step length is the global minimizer.

- 函数应当有充分的下降
- 步长不宜太小

步长

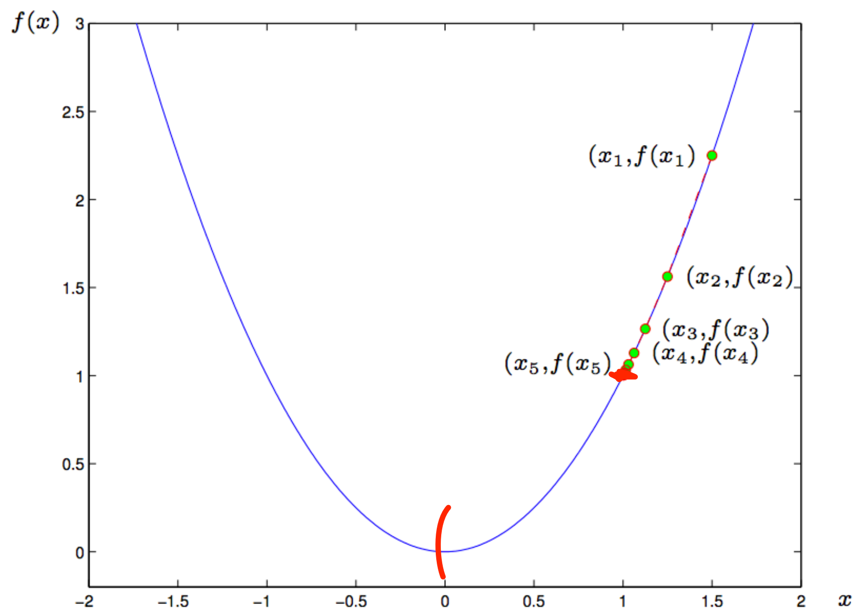
STEPS MIGHT BE TOO LONG



The objective function $f(x) = x^2$ and the iterates $x_{k+1} = x_k + \alpha_k p_k$ generated by the descent directions $p_k = (-1)^{k+1}$ and steps $\alpha_k = 2 + 3/2^{k+1}$ from $x_0 = 2$

步长

STEPS MIGHT BE TOO SHORT



$\alpha_k \to 0$

The objective function $f(x) = x^2$ and the iterates $x_{k+1} = x_k + \alpha_k p_k$ generated by the descent directions $p_k = -1$ and steps $\alpha_k = 1/2^{k+1}$ from $x_0 = 2$

Armijo conditions

首先，我们希望目标函数值能够有足够的下降（sufficient decrease）

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

$\nabla f_k^T p_k < 0$

常数$c_1 \in (0,1)$。将上述条件称为Armijo条件，不等式的右边记为$l(\alpha)$，是一个斜率为$c_1 \nabla f_k^T p_k < 0$的线性函数，从而Armijo条件意味着

$$\phi(\alpha) \le l(\alpha)$$

实际应用中，我们一般将c_1取得很小，例如$c_1 = 10^{-4}$.

Armijo conditions

$\phi(\alpha)$ $l(\alpha)$ $c_1 = 10^{-4}$

$= f_k + c_1 \alpha \nabla f_k^T p_k$

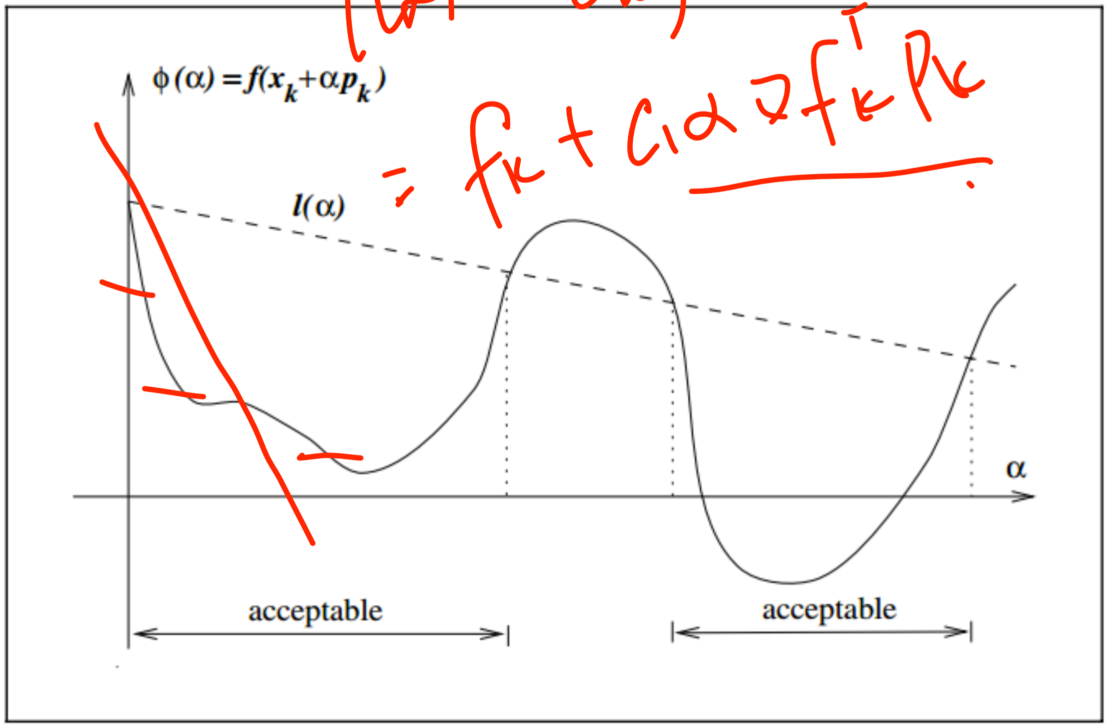


Figure 3.3 Sufficient decrease condition.

curvature condition

然而，当α取得很小的时候，Armijo条件成立，但是这并不是理想的步长，所以我们引入curvature condition，即

$f(x_{k+1})$

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

常数$c_2 \in (c_1, 1)$。不等式的左边恰好是$\phi'(\alpha)$，那么上述不等式意味着$\phi'(\alpha)$应该要大于c_2倍$\phi'(0)$。

$\phi'(\alpha) \geq c_2 \phi'(0)$

$c_2 = 0.9$

when search direction is chosen by Newton or quasi − Newton

$c_2 = 0.1$

when search direction is obtained from nonlinear conjugate grdient method

curvature condition

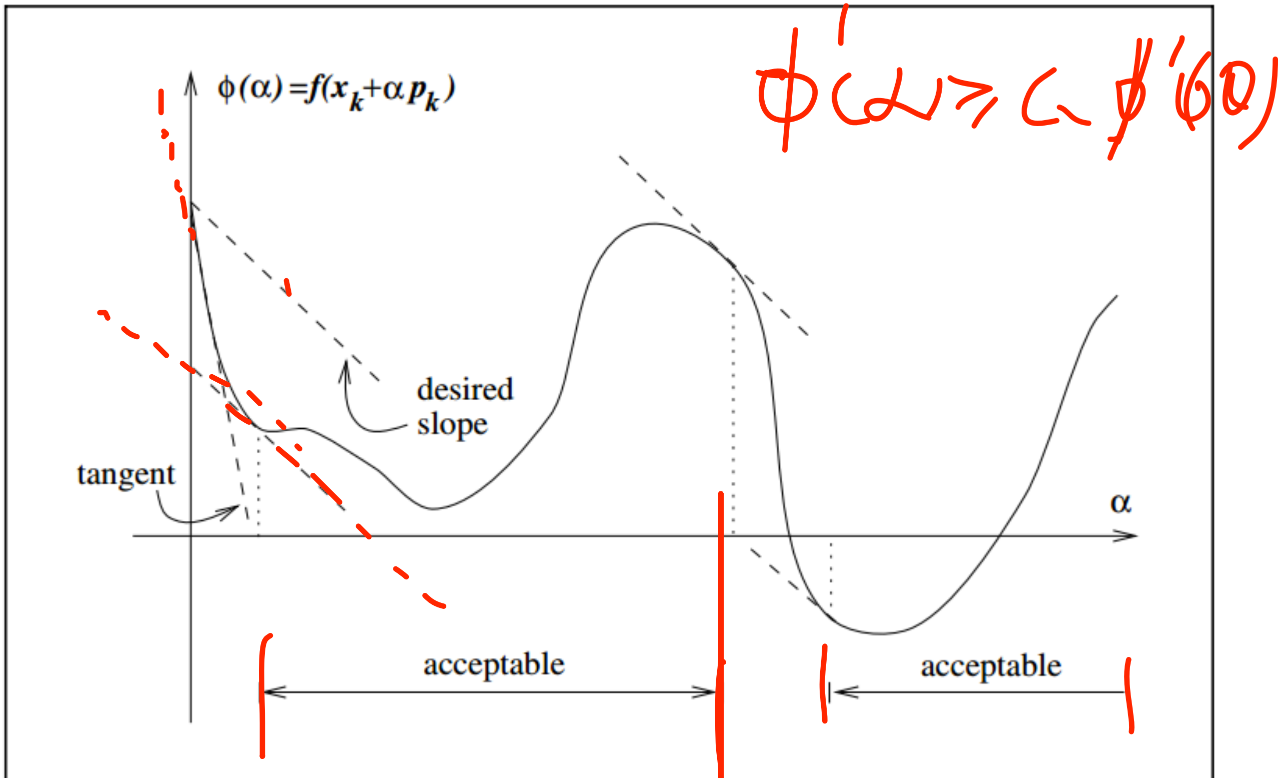


Figure 3.4 The curvature condition.

Wolfe conditions

The sufficient decrease and curvature conditions are known collectively as the *Wolfe conditions*. We illustrate them in Figure 3.5 and restate them here for future reference:

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \tag{3.6a}$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \ge c_2 \nabla f_k^T p_k, \tag{3.6b}$$

with $0 < c_1 < c_2 < 1$.

It is not difficult to prove that there exist step lengths that satisfy the Wolfe conditions for every function f that is smooth and bounded below.

a "loose" line search $c_1 = 10^{-4}$ and $c_2 = 0.9$

Wolfe conditions

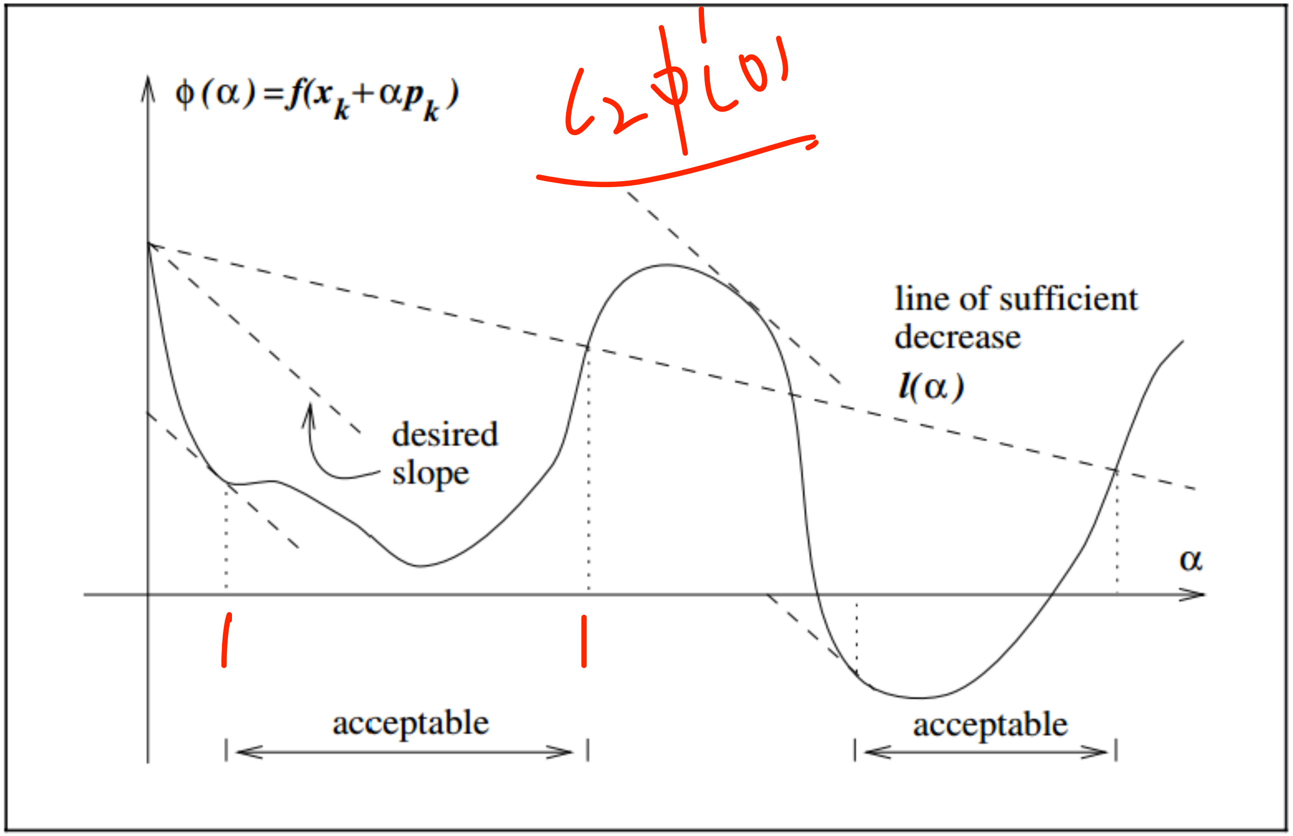


Figure 3.5 Step lengths satisfying the Wolfe conditions.

Wolfe conditions

Algorithm 3.5 (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 $i > 1]$
    $\alpha_* \leftarrow$**zoom**(α_{i-1}, α_i) and **stop**;
  Evaluate $\phi'(\alpha_i)$;
  if $|\phi'(\alpha_i)| \leq -c_2 \phi'(0)$
    set $\alpha_* \leftarrow \alpha_i$ and **stop**;
  if $\phi'(\alpha_i) \geq 0$
    set $\alpha_* \leftarrow$**zoom**(α_i, α_{i-1}) and **stop**;
  Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$;
  $i \leftarrow i + 1$;
end (repeat)

Wolfe conditions

Algorithm 3.6 (zoom).

repeat

Interpolate (using quadratic, cubic, or bisection) 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 $\phi(\alpha_j) \geq \phi(\alpha_{\mathrm{lo}})$
$\alpha_{\mathrm{hi}} \leftarrow \alpha_j$;
else
Evaluate $\phi'(\alpha_j)$;
if $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$
Set $\alpha_* \leftarrow \alpha_j$ and **stop**;
if $\phi'(\alpha_j)(\alpha_{\mathrm{hi}} - \alpha_{\mathrm{lo}}) \geq 0$
$\alpha_{\mathrm{hi}} \leftarrow \alpha_{\mathrm{lo}}$;
$\alpha_{\mathrm{lo}} \leftarrow \alpha_j$;

end (repeat)

Goldstein conditions

与Wolfe conditions类似，Goldstein conditions也是一种可以保证sufficient decrease并且步长α 不会太小的方法，即

$$f(x_k) + (1-c)\alpha \nabla f_k^T p_k \leq f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$$

其中$0 < c < 0.5$

A disadvantage of the Goldstein conditions vis-à-vis the Wolfe conditions is that the first inequality in (3.11) may exclude all minimizers 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.

Goldstein conditions

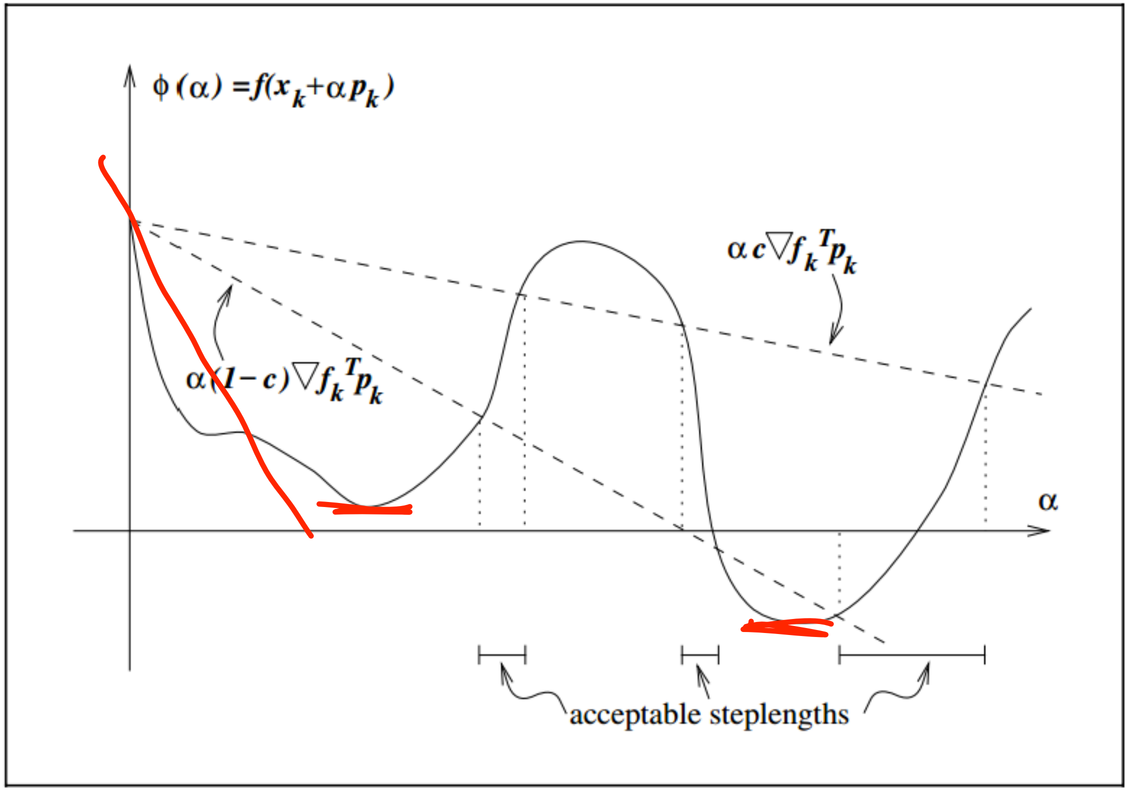


Figure 3.6 The Goldstein conditions.

Backtracking line search

在实际应用中，我们为了提高效率，会选在抛弃掉Wolfe conditions的第二个条件，只使用sufficient decrease conditions来求取最优步长，常用的方法为backtracking methods。

Algorithm 3.1 (Backtracking Line Search).

Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;
repeat until $f(x_k + \alpha p_k) \le f(x_k) + c\alpha \nabla f_k^T p_k$
$\alpha \leftarrow \rho\alpha$;
end (repeat)
Terminate with $\alpha_k = \alpha$.

α
ℓα
ℓα

一般而言，在Newton's method和quasi-Newton's method中我们选择$\bar{\alpha} = 1$，在steepest descent和conjugate gradient方法中我们可能会有其他的选择。ρ的取值往往随循环数的变化而变化，我们需要保证$\rho \in [\rho_{lo}, \rho_{hi}], 0 < \rho_{lo} < \rho_{hi} < 1$.

Steepest descent

在最速下降法（steepest descent method）中，线搜索方向为

$$p_k = -\nabla f_k$$

Wolfe

Backtracking

故主要的迭代公式为

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

2 Steepest descent

2.1 The algorithm

Step 0 Given $\boldsymbol{x}^0$, set $k := 0$.

Step 1 $\boldsymbol{d}^k := -\nabla f(\boldsymbol{x}^k)$. If $\|\boldsymbol{d}^k\| \leq \epsilon$, then stop.

$\|\nabla f_k\| \leq \varepsilon$

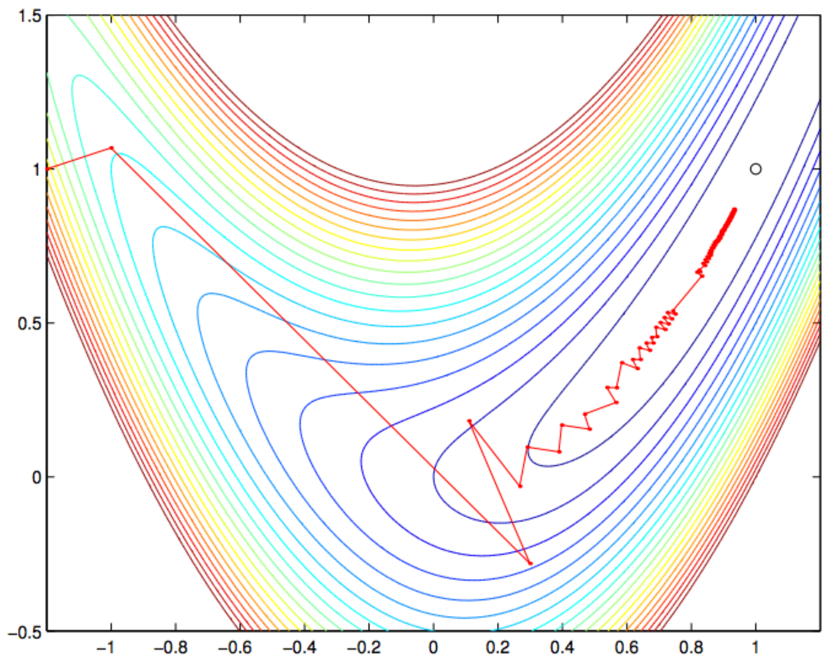
$\nabla f(x_k) \simeq 0$

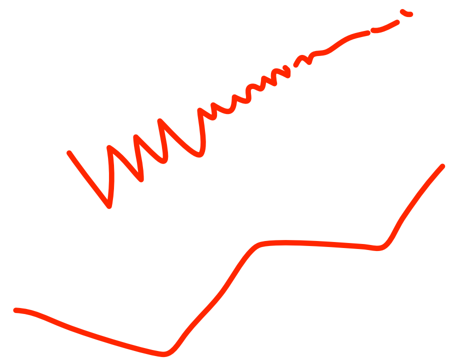
Step 2 Solve $\min_\lambda h(\lambda) := f(\boldsymbol{x}^k + \lambda \boldsymbol{d}^k)$ for the step-length λ^k, perhaps chosen by an exact or inexact line-search.

Step 3 Set $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \lambda^k \boldsymbol{d}^k$, $k \leftarrow k + 1$. Go to **Step 1**.

Steepest descent

STEEPEST DESCENT EXAMPLE





Contours for the objective function $f(x, y) = 10(y - x^2)^2 + (x - 1)^2$, and the iterates generated by the Generic Linesearch steepest-descent method

Steepest descent

Advantages and disadvantages of steepest descent:

- $\oplus$ Globally convergent (converges to a local minimiser from any starting point x^0).
- $\oplus$ Many other methods switch to steepest descent when they do not make sufficient progress.
- $\ominus$ Not scale invariant (changing the inner product on $\mathbb{R}^n$ changes the notion of gradient!).
- $\ominus$ Convergence is usually very (very!) slow (linear).
- $\ominus$ Numerically, it is often not convergent at all.

Newton’s method

在牛顿法（Newton’s method）中，线搜索方向为

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

这是因为考虑f在x_k附近的二阶泰勒展开近似，从而有

$$f(x) \approx f(x_k) + \nabla f_k'(x - x_k) + \frac{1}{2}(x - x_k)'\nabla^2 f(x_k)(x - x_k)$$

对上式右边求最小值，从而

x_{k+1}

$$\nabla f_k' + \nabla^2 f(x_k)(x - x_k) = 0$$

从而，主要的迭代公式为

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1}\nabla f_k$$

Newton's method

The algorithm

Step 0 Given $\boldsymbol{x}^0$, set $k := 0$.

Step 1 $\boldsymbol{d}^k := -(\nabla^2 f(\boldsymbol{x}^k))^{-1}\nabla f(\boldsymbol{x}^k)$.

If $||\boldsymbol{d}^k|| \leq \epsilon$, then stop.

Step 2 Set $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \boldsymbol{d}^k$, $k \leftarrow k+1$.

Go to **Step 1**.

λ

$\lambda_k \leftarrow$

Comments

- The method assumes that $\nabla^2 f(\boldsymbol{x}^k)$ is nonsingular at each iteration
- There is no guaranteee that $f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k)$
- We can augment the algorithm with a line-search:

$$\min_{\lambda} f(\boldsymbol{x}^k + \lambda\boldsymbol{d}^k)$$

$f(x) \approx$ 二阶 Taylor

Newton's method

收敛性：局部二阶收敛

Theorem 5 (Local Convergence of Newton's Method). *Let the Hessian H of $f \in C^2$ be uniformly Lipschitz continuous on $\mathbb{R}^n$. Let iterates x^k be generated via the Generic LSM with B-A steps using $\alpha_{\text{init}} = 1$ and $\beta < \frac{1}{2}$, and using the Newton search direction n^k, defined by $H^k n^k = -g^k$. If $(x^k)_{\mathbb{N}}$ has an accumulation point x^* where $H(x^*) \succ 0$ (positive definite) then*

i) $\alpha^k = 1$ *for all k large enough,*

ii) $\lim_{k\to\infty} x^k = x^*$,

iii) the sequence converges Q-quadratically, that is, there exists $\kappa > 0$ such that

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} \leq \kappa.$$

$x^0 \to x^*$

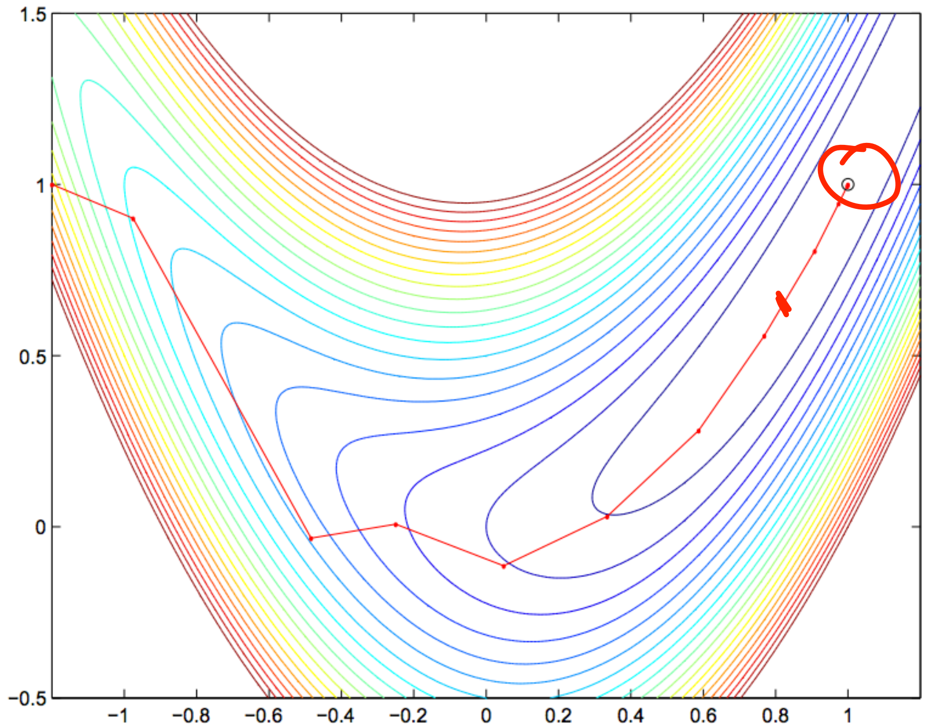
The mechanism that makes Theorem 5 work is that once the sequence $(x^k)_{\mathbb{N}}$ enters a certain domain of attraction of x^*, it cannot escape again and quadratic convergence to x^* commences.

二阶

Note that this is only a local convergence result, that is, Newton's method is not guaranteed to converge to a local minimiser from all starting points.

Newton's method

The fast convergence of Newton's method becomes apparent when we apply it to the Rosenbrock function:



Contours for the objective function $f(x,y) = 10(y - x^2)^2 + (x-1)^2$, and the iterates generated by the Generic Linesearch Newton method.

Newton’s method

如果海色矩阵$\nabla^2 f(x_k)$不正定，那么牛顿法可能出现迭代方向并不是下降方向的情况，此时我们需要考虑从将海瑟矩阵修正成一个正定矩阵，算法框架如下：

B_k $\nabla^2 f_k + E_k$

Algorithm 3.2 (Line Search Newton with Modification).

Given initial point x_0;

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

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

$H+\tau I \succ 0$

一种修正方法如下：

$$\nabla^2 f(x_k)+\tau I$$

$\tau > 0$

Algorithm 3.3 (Cholesky with Added Multiple of the Identity).

Choose $\beta > 0$;
if $\min_i a_{ii} > 0$
  set $\tau_0 \leftarrow 0$;
else
  $\tau_0 = -\min(a_{ii}) + \beta$;
end (if)
for $k = 0, 1, 2, \ldots$
  Attempt to apply the Cholesky algorithm to obtain $LL^T = A + \tau_k I$;
  if the factorization is completed successfully
    stop and return L;
  else
    $\tau_{k+1} \leftarrow \max(2\tau_k, \beta)$;
  end (if)
end (for)

$\beta = 10^{-3}$

共轭梯度法

共轭梯度法(conjugate gradient method) 是一种用于求解线性方程组的迭代方法，记线性方程组为

$$Ax = b$$

LU $O(n^3)$

$A \succeq 0$ Cholesky

那么上述问题等价于求解优化问题

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

我们引入记号

$$r(x) = \nabla\phi(x) = Ax - b$$

$Ax=b$ x^*

当$x = x_k$时，记

$$r_k = Ax_k - b$$

x_0 x_1 ... x_k

共轭方向法

我们称一系列非零向量$\{p_0, p_1, \ldots, p_l\}$是关于正定矩阵A共轭的，如果他们满足：

$$p_i^T A p_j = 0, for\ all\ i \neq j$$

I

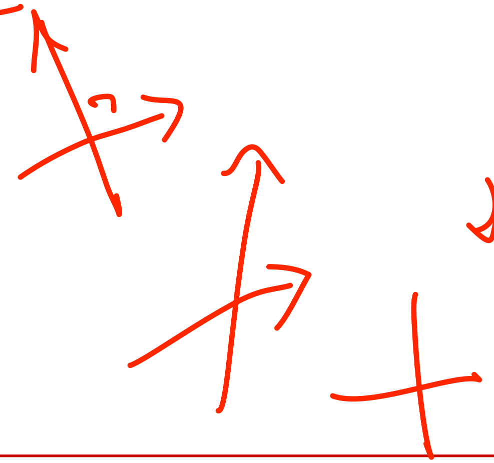
Ap_i

$A = I$

$p_i^T p_j = 0$

显然共轭的向量组也是线性无关的向量组。

$k_1 p_1 + \cdots + k_l p_l = 0$



共轭方向法

共轭方向法是指对于给定的点$x_0 \in \mathbb{R}^n$，一系列关于正定矩阵A共轭的非零向量组$\{p_0, p_1, \dots, p_l\}$，那么我们生成序列

$x_0\ x_1 \cdots x_k$

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

其中α_k是使得函数$\phi(x) = \frac{1}{2}x^T A x - b^T x$最小化的步长，所以

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

那么可以证明，对于任意的按照上述方式生成的序列$\{x_k\}$,该序列收敛到x^*是线性方程组$Ax = b$的解，并且上述算法最多在n步收敛。

共轭方向法

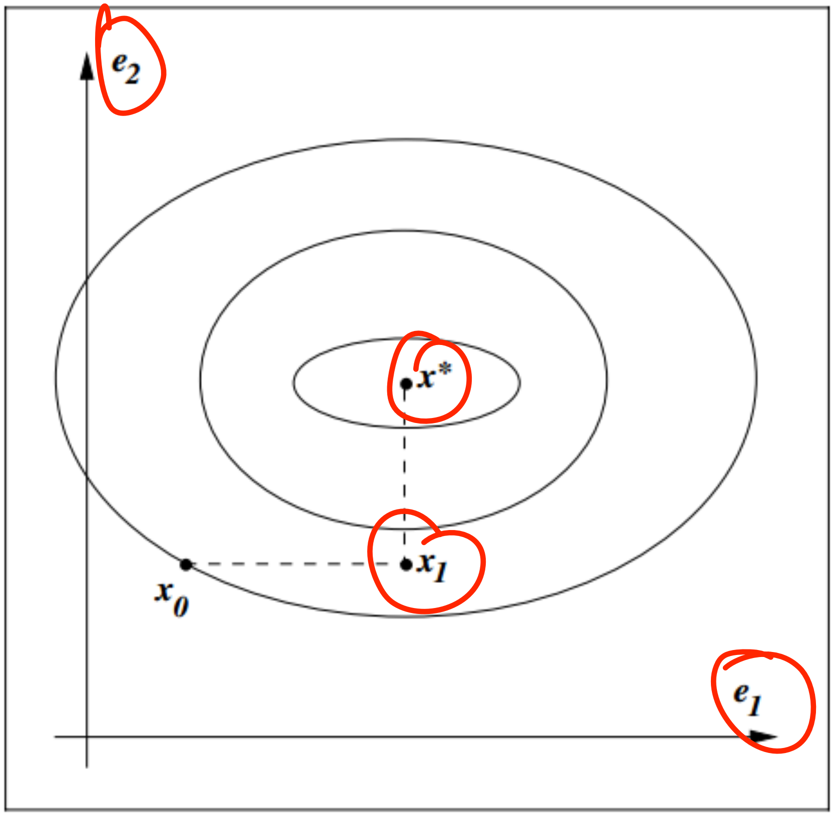


Figure 5.1 Successive minimizations along the coordinate directions find the minimizer of a quadratic with a diagonal Hessian in n iterations.

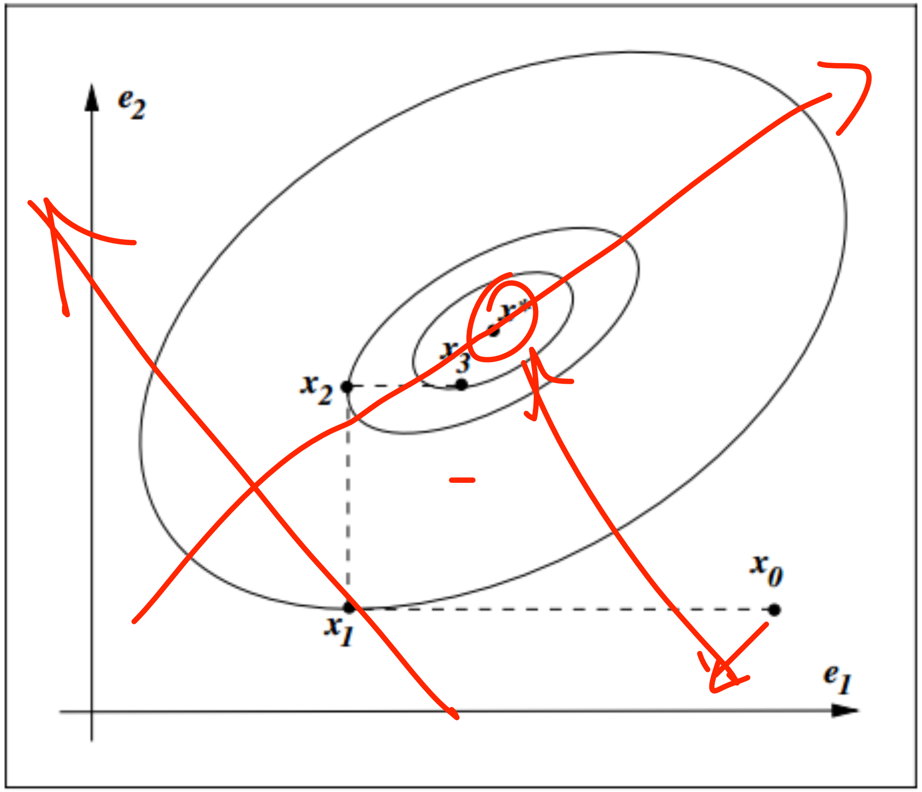
$x_0 \quad x_1 \quad x_2$

n

A 对角

共轭方向法

$S^T A S = D$



$$\hat{x} = S^{-1} x,$$

$x = S\hat{x}$

$$\hat{\phi}(\hat{x}) \stackrel{\text{def}}{=} \phi(S\hat{x}) = \frac{1}{2}\hat{x}^T (S^T A S)\hat{x} - (S^T b)^T \hat{x}.$$

对角

Figure 5.2 Successive minimization along coordinate axes does not find the solution in n iterations, for a general convex quadratic.

共轭梯度法

α_k 问题的关键是选择一系列关于正定矩阵A共轭的非零向量组 $\{p_0, p_1, \dots, p_l\}$，共轭梯度法最巧合的是每一个方向 p_k 仅与前一个方向 p_{k-1} 相关，即假设：

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

β_k 满足条件使得 p_k 与 p_{k-1} 关于矩阵A共轭，从而解出

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

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

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

$p_{k-1}^T A r_k =$

$p_{k-1}^T A(-r_k + \beta_k p_{k-1}) = 0$

共轭梯度法

Algorithm (CG–Preliminary Version).

Given x_0;

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

负梯度.

while $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k};$$

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

$$r_{k+1} \leftarrow A x_{k+1} - b;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end (while)

n.步收敛

$Ax=b$

共轭梯度法

Algorithm (CG).

Given x_0;

Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k};$$

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

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end (while)

运行结束.

α_k

β_k

共轭梯度法

Remark:

n　　$Ax=b$

- 共轭梯度法更适合数据容量比较大的情况。一般数据容量的情况下高斯消元法或者Cholesky分解等方法更适合。

$O(n^3)$

- 可以证明，如果A有r个不同的特征值，共轭梯度法最多在r步内收敛。

B　n

共轭梯度法

Algorithm 5.4 (FR).

Given x_0;

Evaluate $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$;

Set $p_0 \leftarrow -\nabla f_0$, $k \leftarrow 0$;

while $\nabla f_k \neq 0$

  Compute α_k and set $x_{k+1} = x_k + \alpha_k p_k$;

  Evaluate ∇f_{k+1};

$$\beta_{k+1}^{\mathrm{FR}} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k};$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{FR}} p_k;$$

$$k \leftarrow k + 1;$$

end (while)

$\min\limits_{x \in R} f(x) \sim f_k + \nabla f_k (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f (x - x_k)$

Wolf

α_k

β_k

拟牛顿法

BFGS

拟牛顿法（quasi-Newton method）是基于牛顿法基础上的一系列改进方法，对于普通牛顿法而言

$$p_k = -\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

- 需要涉及到矩阵求逆
- 海瑟矩阵不正定

x_k x^*

拟牛顿法的基本思路是估计海瑟矩阵或者海瑟矩阵的逆。

拟牛顿法

在迭代的点x_k处用函数的二阶泰勒展开近似函数，得到

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

其中B_k是海瑟矩阵的一个估计，我们需要在每一轮迭代中对其进行更新，根据牛顿法，搜索方向p_k可以表示为

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

下一个状态为

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

对于下一个状态，我们同样可以用二阶泰勒展开去近似函数：

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$$

拟牛顿法

x_k m_k m_{k+1}

我们希望这两个近似具有一定的相似性，所以要求

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

移项得到

x_k x_{k+1}

$$\alpha_k B_{k+1} p_k = \nabla f_{k+1} - \nabla f_k$$

令

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

那么上式可以简记为

x_{k+1}

$$B_{k+1} s_k = y_k$$

如果记$H_{\mathrm{k}} = B_k^{-1}$,那么$H_{\mathrm{k}}$估计的是海瑟矩阵的逆，并且满足关系

$$H_{\mathrm{k+1}} y_k = s_k$$

拟牛顿法

x_{k+1} x_k $H_{k+1} = H_k +$

m_k m_{k+1}

我们希望$H_{\mathrm{k+1}}$与H_{k}是接近的，因为这两个近似是相近的，所以$H_{\mathrm{k+1}}$应当满足条件

$$\min_{H} ||H - H_k||$$

$$s.t. H = H^T, H y_k = s_k$$

可以求出，$H_{\mathrm{k+1}}$与H_{k}应当满足下述关系

$$H_{k+1} = \left(1 - \rho_k s_k y_k^T\right) H_k \left(1 - \rho_k y_k s_k^T\right) + \rho_k s_k s_k^T$$

其中$\rho_{\mathrm{k}} = \frac{1}{y_k^T s_k}$,上述更新公式称为BFGS更新公式。

拟牛顿法

Algorithm 6.1 (BFGS Method).

Given starting point x_0, convergence tolerance $\epsilon > 0$,
inverse Hessian approximation H_0;

$k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$;

Compute search direction

$$p_k = -H_k \nabla f_k;$$

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions (3.6);

Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;

Compute H_{k+1}

$k \leftarrow k + 1$;

end (while)

拟牛顿法

Global result

if f is strongly convex, BFGS with backtracking line search (EE236B, lecture 10-6) converges from any $x^{(0)}$, $H_0 \succ 0$

Local convergence

if f is strongly convex and $\nabla^2 f(x)$ is Lipschitz continuous, local convergence is *superlinear*: for sufficiently large k,

$$\|x^{(k+1)} - x^\star\|_2 \leq c_k \|x^{(k)} - x^\star\|_2 \to 0$$

where $c_k \to 0$

(*cf.*, quadratic local convergence of Newton method)

我们在这里

- 问题答疑：http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象
- 新浪微博：ChinaHadoop

法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

约束最优化问题

~~KKT~~

svm

约束最优化（constrained optimization）问题的标准形式为：

$$\min_{x\in\mathbb{R}^n} f(x) \ \ s.t. \begin{cases} c_i(x) = 0, i \in \mathcal{E} \\ c_i(x) \geq 0, i \in \mathcal{I} \end{cases}$$

其中f和c_i是光滑的实值函数，$\mathcal{E}$和$\mathcal{I}$是两个指标集，我们称$f(x)$为目标函数（objective function），称$c_i(x) = 0, i \in \mathcal{E}$为等式约束（equality constraints），称$c_i(x) \geq 0, i \in \mathcal{I}$为不等式约束（inequality constraints）。我们定义满足所有约束条件的x为可行点（feasible point），所有可行点组成的集合为可行域（feasible set），记为：

min f(x)

$$\Omega = \{c_i(x) = 0, i \in \mathcal{E}, c_i(x) \geq 0, i \in \mathcal{I}\}$$

min f(x)

约束最优化问题

定义（激活集）　　所有等式约束和不等式约束取等的情况为激活集(active set)，记为

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} | c_i(x) = 0\}$$

$$\begin{cases} 2x+y=0 \\ 4x+2y=0 \end{cases}$$

定义（LICQ条件）　　对于点x和激活集$\mathcal{A}(x)$，如果激活集的梯度向量，即

$$\begin{pmatrix}2\\1\end{pmatrix} \quad \begin{pmatrix}4\\2\end{pmatrix}$$

$$\{\nabla c_i(x), i \in \mathcal{A}(x)\}$$

线性无关，那么称LICQ(linear independence constraint qualification)条件成立.

KKT条件

定义约束优化问题的拉格朗日函数为

$$\mathcal{L}(x,\lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$

$>$

$\leq$

KKT条件

Theorem 12.1 (First-Order Necessary Conditions).

Suppose that x^* is a local solution of (12.1), that the functions f and c_i in (12.1) are continuously differentiable, and that the LICQ holds at x^*. Then there is a Lagrange multiplier vector λ^*, with components λ_i^*, $i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions are satisfied at (x^*, λ^*)

$$
\begin{aligned}
\nabla_x \mathcal{L}(x^*, \lambda^*) &= 0, && \\
c_i(x^*) &= 0, && \text{for all } i \in \mathcal{E}, \\
c_i(x^*) &\geq 0, && \text{for all } i \in \mathcal{I}, \\
\lambda_i^* &\geq 0, && \text{for all } i \in \mathcal{I}, \\
\lambda_i^* c_i(x^*) &= 0, && \text{for all } i \in \mathcal{E} \cup \mathcal{I}.
\end{aligned}
$$

于

$c_i(x^*) > 0$

x^* $A(x)$

称最后一个条件为互补性条件，它说明约束优化问题的解，或者λ_i^*等于零，或者这个点是激活集中的点。

KKT条件

$$\min_x \left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^4 \qquad \text{s.t.} \qquad \begin{bmatrix} 1 - x_1 - x_2 \\ 1 - x_1 + x_2 \\ 1 + x_1 - x_2 \\ 1 + x_1 + x_2 \end{bmatrix} \geq 0.$$

active

It is fairly clear from Figure 12.11 that the solution is $x^* = (1, 0)^T$. The first and second constraints in (12.36) are active at this point. Denoting them by c_1 and c_2 (and the inactive constraints by c_3 and c_4), we have

λ_i^* $c_i(x^*)$

$$\nabla f(x^*) = \begin{bmatrix} -1 \\ 1 \\ -\frac{1}{2} \end{bmatrix}, \qquad \nabla c_1(x^*) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \qquad \nabla c_2(x^*) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Therefore, the KKT conditions (12.34a)–(12.34e) are satisfied when we set

$$\lambda^* = \left(\frac{3}{4}, \frac{1}{4}, 0, 0\right)^T.$$

KKT条件

Q：对于一个最优解x^*而言，λ_i^*的意义是什么？

假设$x^*(\epsilon)$是对x^*的一个扰动，那么

$$
\begin{aligned}
-\epsilon\|\nabla c_i(x^*)\| = c_i(x^*(\epsilon)) - c_i(x^*) &\approx (x^*(\epsilon) - x^*)^T \nabla c_i(x^*), \\
0 = c_j(x^*(\epsilon)) - c_j(x^*) &\approx (x^*(\epsilon) - x^*)^T \nabla c_j(x^*), \\
&\text{for all } j \in \mathcal{A}(x^*) \text{ with } j \neq i.
\end{aligned}
$$

$$
\begin{aligned}
f(x^*(\epsilon)) - f(x^*) &\approx (x^*(\epsilon) - x^*)^T \nabla f(x^*) \\
&= \sum_{j \in \mathcal{A}(x^*)} \lambda_j^* (x^*(\epsilon) - x^*)^T \nabla c_j(x^*) \\
&\approx -\epsilon \|\nabla c_i(x^*)\| \lambda_i^*.
\end{aligned}
$$

By taking limits, we see that the family of solutions $x^*(\epsilon)$ satisfies

$$
\frac{df(x^*(\epsilon))}{d\epsilon} = -\lambda_i^* \|\nabla c_i(x^*)\|.
$$

对偶问题

考虑只有不等式约束的情况：

P D

$$\min_{x\in\mathbb{R}^n} f(x) \;\; s.t. \;\; c_i(x) \geq 0, i \in 1,2,\ldots,m$$

令

$$c(x) \stackrel{\text{def}}{=} (c_1(x), c_2(x), \ldots, c_m(x))^T$$

那么问题简写为

$$\min_{x\in\mathbb{R}^n} f(x) \;\; s.t. \;\; c(x) \geq 0$$

7

记其拉格朗日函数为

$$\mathcal{L}(x,\lambda) = f(x) - \lambda^T c(x)$$

对偶问题

定义对偶问题的目标函数为

$$q(\lambda) \overset{\text{def}}{=} \inf_{x} \mathcal{L}(x,\lambda) = \inf_{x} f(x) - \lambda^T c(x)$$

其中$q(\lambda)$ 的定义域为

$$D \overset{\text{def}}{=} \{\lambda | q(\lambda) > -\infty\}$$

从而原优化问题的对偶问题为

$$\max_{\lambda \in \mathbb{R}^n} q(\lambda) \quad s.t. \ \lambda \geq 0$$

λ^*

对偶问题

考虑问题

$$\min_{(x_1,x_2)} 0.5(x_1^2 + x_2^2) \quad s.t. x_1 - 1 \geq 0$$

$\begin{cases} x_1 = 1 \\ x_2 = 0 \end{cases}$

$0.5x_1^2 - x_1$

$x_1(0.5x_1 - 1)$

$0 \quad 2$

$x_1 = 2$

$L(x_1, x_2, \lambda) = 0.5(x_1^2 + x_2^2) - \lambda(x_1 - 1)$

$g(\lambda) = \inf_{(x_1, x_2)} L(x_1, x_2, \lambda) = 0.5\lambda^2 - \lambda^2 + \lambda$

$= \lambda - 0.5\lambda^2$

$\frac{\partial L}{\partial x_1} = x_1 - \lambda = 0 \quad \frac{\partial L}{\partial x_2} = x_2 = 0$

$\lambda(1 - 0.5\lambda)$

$\max \lambda - 0.5\lambda^2 \quad \lambda \geq 0$

$\lambda = 1$ 极大

D.

对偶问题

线性规划（linear programming）

P: $\min c^T x \ \ s.t. \ Ax - b \geq 0$

D: $\max b^T \lambda \ \ s.t. \ A\lambda = c, \lambda \geq 0$

$$L(x, \lambda) = c^T x - \lambda^T (Ax - b)$$

$$= (c^T - \lambda^T A)x + \lambda^T b$$

$$g(\lambda) > -\infty$$

$$c^T - \lambda^T A = 0$$

$$g(\lambda) = \lambda^T b \quad s.t. \quad c^T - \lambda^T A = 0 \quad \lambda \geq 0$$

我们在这里

- 问题答疑：
  http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象
- 新浪微博：ChinaHadoop