运筹学

11. 非线性规划的搜索算法

李 力
清华大学

Email: li-li@tsinghua.edu.cn

2023.11.

迭代搜索算法

迭代求解的基本想法

为了求某可微函数(假定无约束)的最优解，根据前面的叙述，可如下进行：

令该函数的梯度等于零，由此求得局部极值点；然后用充分条件进行判别，是否已经求出需要得到的解。

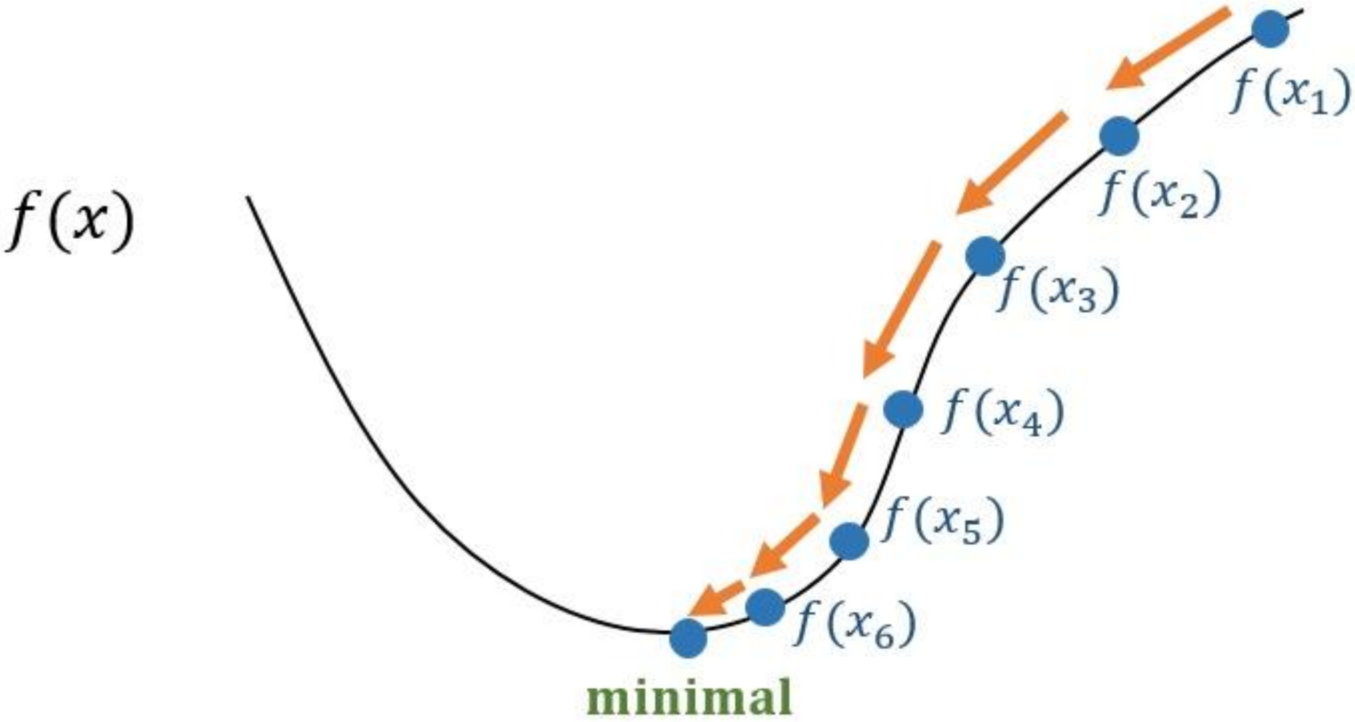
对某些较简单的函数，这样做有时是可行的；但对一般n元函数f(X)来说，由条件f(X)=0得到的常是一个非线性方程组，解它相当困难。对于不可微函数，当然谈不上使用这样的方法。

为此，常直接使用迭代法，将求解转化为不断改变方向和步长的搜索过程。

迭代算法关注两点：前进方向和前进的步长

迭代求解的基本想法

$$X_{k+1} = X_k + t_k D_k$$



可行下降方向

对于优化问题 $\min\limits_{X\in\Omega} f(X)$ ，给定可行解 $\hat{X}\in\Omega$ 以

及向量 $D\in R^n$ ，如果存在 $\bar{t}>0$ 满足

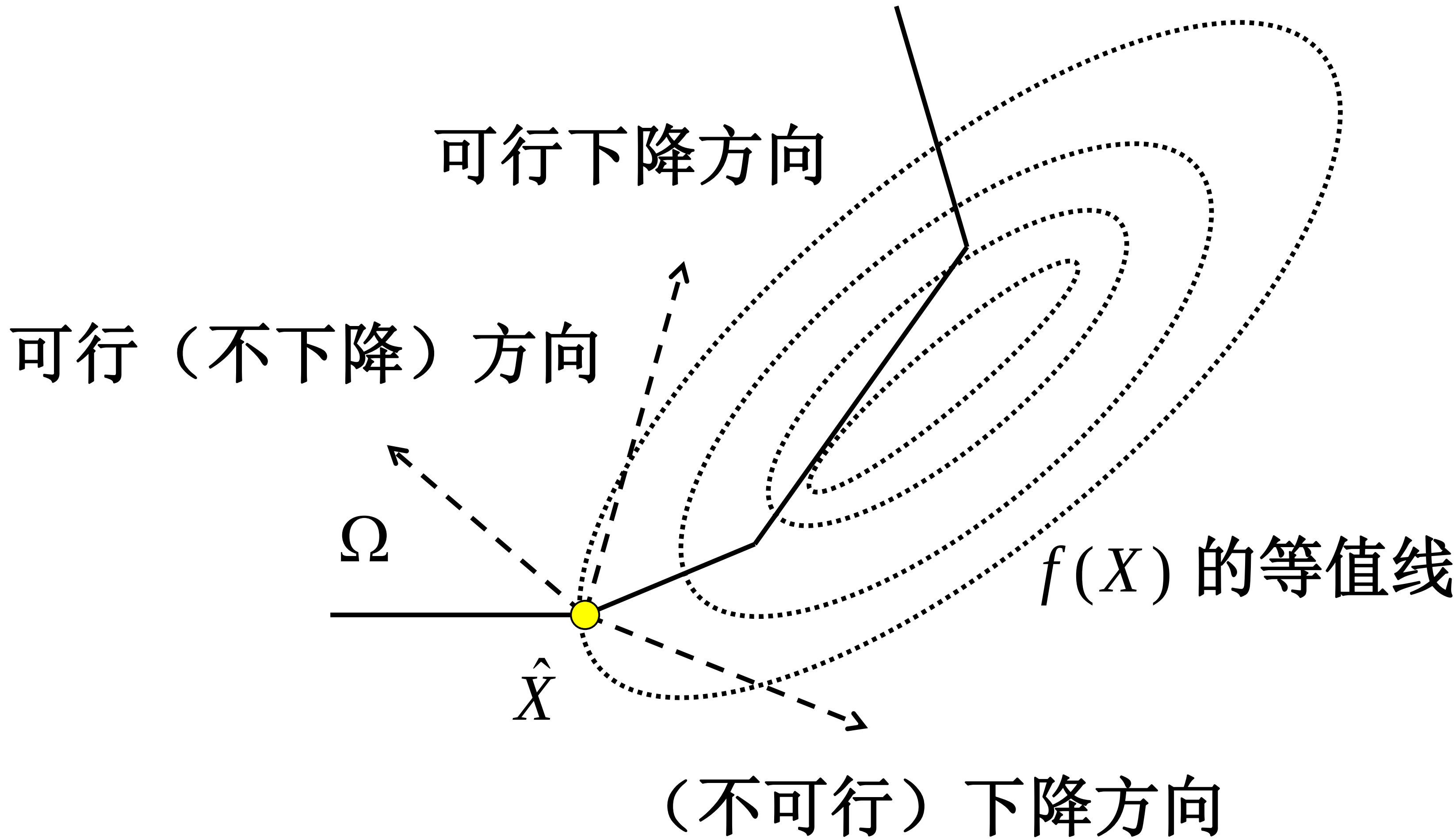
$$\hat{X}+tD\in\Omega,\ \forall 0<t\le\bar{t}$$

称 D 是 $\hat{X}$ 处的可行方向，如果存在 $\bar{t}>0$ 满足

$$f\left(\hat{X}+tD\right)<f(\hat{X}),\ \forall 0<t\le\bar{t}$$

称 D 是 $\hat{X}$ 处的下降方向，既可行又下降的方向

称为可行下降方向


可行下降方向
可行（不下降）方向
Ω
$f(X)$ 的等值线
$\hat{X}$
（不可行）下降方向

可行下降迭代算法

确定初始可行解 $X_0 \in \Omega$ $\rightarrow$ 确定可行下降方向 D_0

$\Rightarrow$ 一维搜索确定 λ_0 满足 $X_1 = X_0 + t_0 D_0 \in \Omega$ 以及

$f(X_1) < f(X_0)$

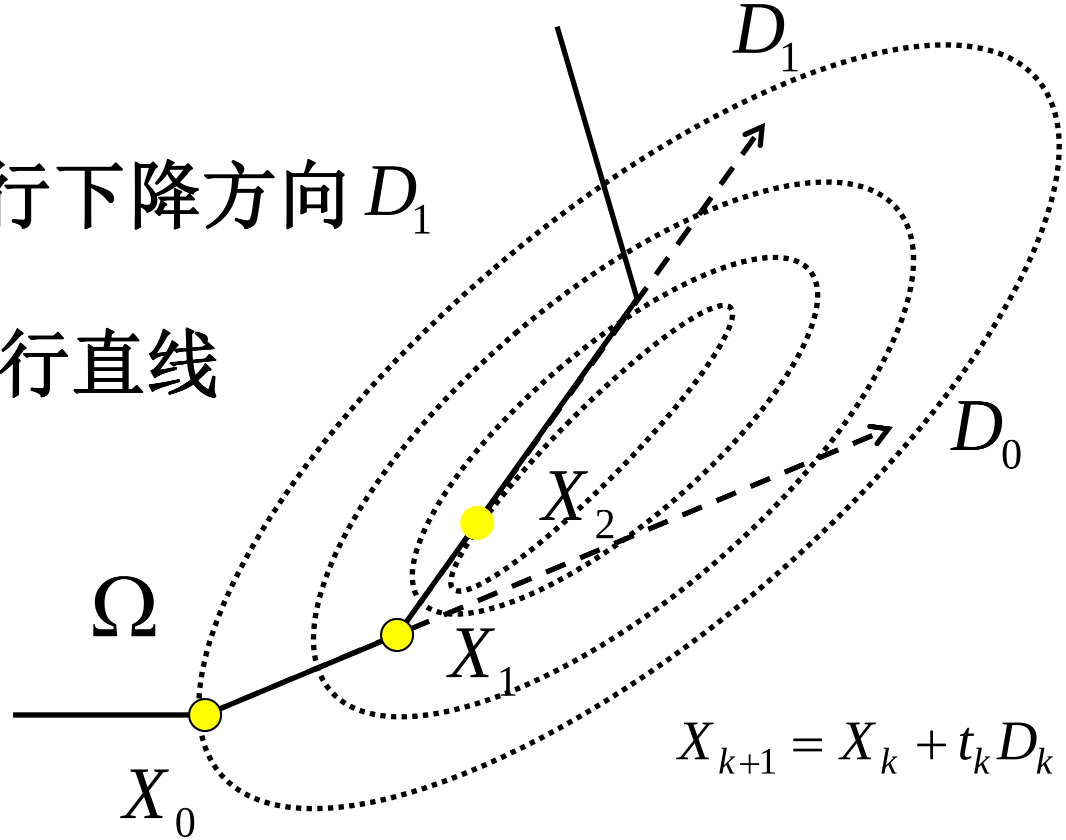
$\Rightarrow$ 确定 X_1 处的可行下降方向 D_1

$\Rightarrow$ 在 X_1 处沿 D_1 进行直线

搜索确定 X_2

$\cdots$

如此继续



凸规划最优解与可行下降方向的关系

对于凸规划 $\min\limits_{X \in \Omega} f(X)$（$\Omega$凸集，$f(X)$凸函数）

$X^* \in \Omega$ 是最优解的充要条件是在该点不存在可行下降方向

必要性显然。充分性反证：如果有另一点更好，连接两点可得可行下降方向。

跟据每步搜索使用的信息可以分类Depending on information of the problem being used to create a new iterate, we have

(a) Zero-order algorithms. 零阶算法 Popular when the gradient and Hessian information are difficult to obtain, e.g., no explicit function forms are given, functions are not differentiable, etc.

(b) First-order algorithms.一阶算法 Most popular now-days, suitable for large scale data optimization with low accuracy requirement, e.g., Machine Learning, Statistical Predictions.

(c) Second-order algorithms.二阶算法 Popular for optimization problems with high accuracy need, e.g., some scientific computing, etc.

一维和一阶不同

一维搜索

条件：已知 $X_0 \in \Omega$ 和可行下降方向 $D \in R^n$

精确搜索：求解单变量优化问题（一维搜索）

$$
\begin{aligned}
\min\ & \varphi(t) = f(X_0 + tD) \\
\text{s.t.}\ & X_0 + tD \in \Omega \\
& t \geq 0
\end{aligned}
$$

非精确搜索：找到一个 $\hat{t} > 0$ 满足

$X_0 + \hat{t}D \in \Omega,\ \varphi(\hat{t}) \leq \varphi(0) + \alpha \hat{t} \varphi'(0)$，并且 $\hat{t}$ 不能太小

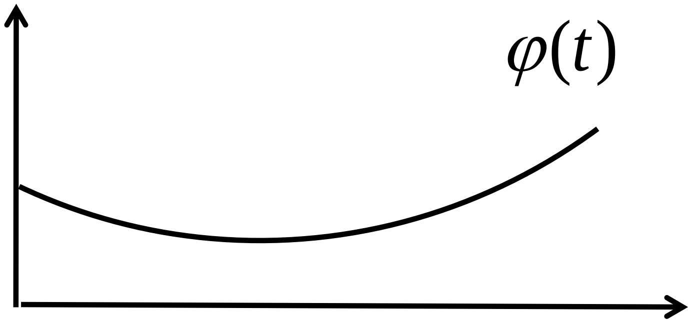
其中 $\alpha > 0$，由于 D 是下降方向，所以 $\varphi'(0) < 0$

精确搜索的基本途经

1）确定初始区间

用试探法确定一个单谷区间

可用步长加倍或减倍的方法



2）逐步压缩区间

按照一定规则在上述区间内选点，通过计算并比较这些点上的函数值逐步缩小包含局部最优解的区间直至区间长度小于给定阈值

单谷/单峰函数

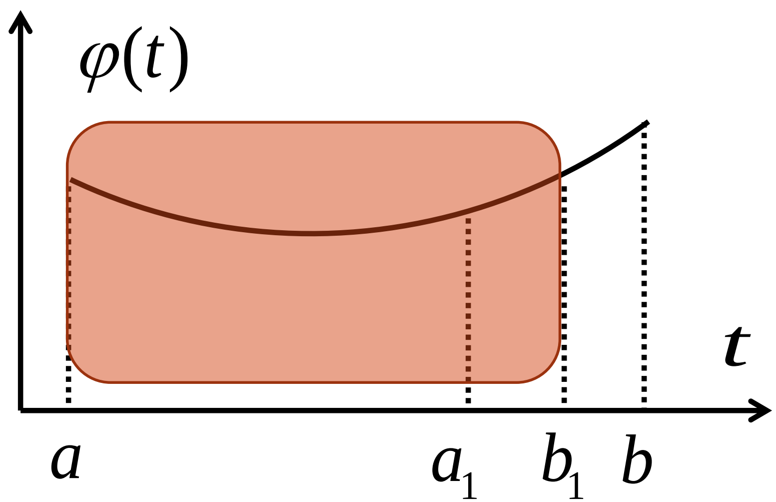
单峰函数是在所考虑的区间中只有一个严格局部极大值(峰值)的实值函数。如果函数f(x)在区间[a, b]上只有唯一的最大值点C，而在最大值点C的左侧，函数单调增加；在点C的右侧，函数单调减少，则称这个函数为区间[a, b]上的单峰函数。

单谷函数是在所考虑的区间中只有一个严格局部极小值(谷值)的实值函数。如果函数f(x)在区间[a, b]上只有唯一的最小值点C，而在最小值点C的左侧，函数单调减少；在点C的右侧，函数单调增加，则称这个函数为区间[a, b]上的单谷函数。

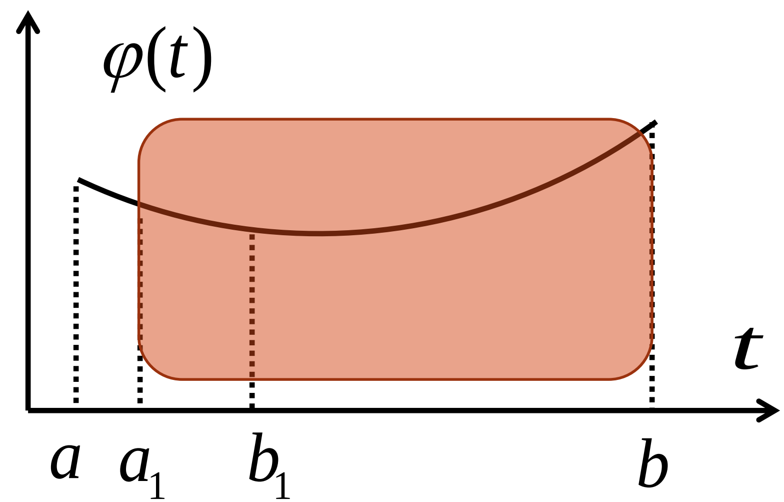
和凸函数的关系？

区间压缩原理

已知闭区间 $[a,b]$ 是单谷区间，在其内部任取两点 $a_1 < b_1$ ，计算 $\varphi(a_1),\varphi(b_1)$ ，如果 $\varphi(a_1) < \varphi(b_1)$ ，局部最优解在区间 $[a,b_1]$，否则局部最优解在区间 $[a_1,b]$ ，两种情况均能压缩区间



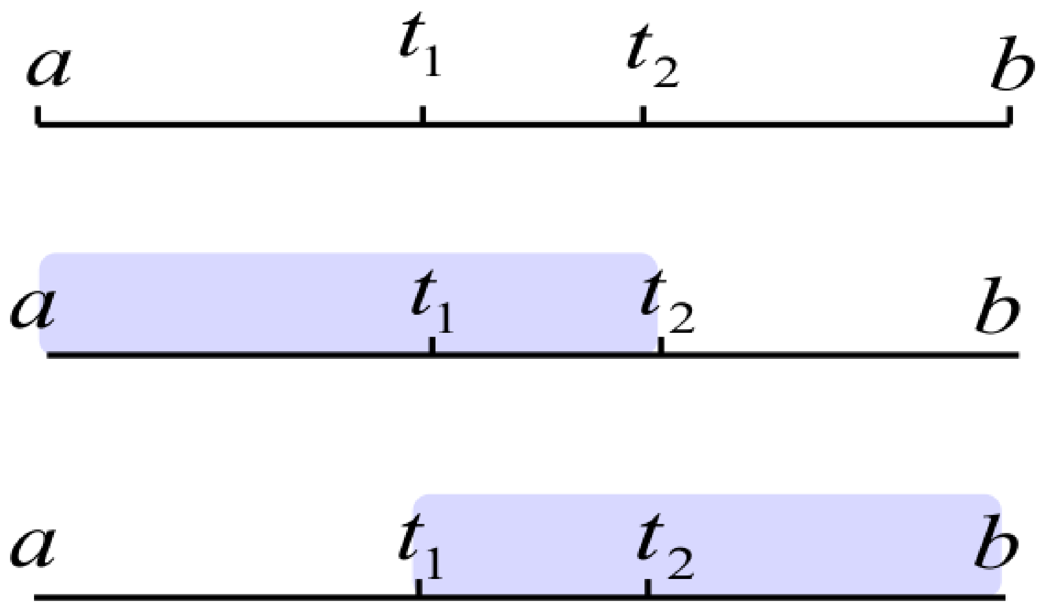
第一种情况



第二种情况

一维搜索的0.618 法的原理

基本想法：每**计算一个函数值**能将区间**压缩一个固定比值** c



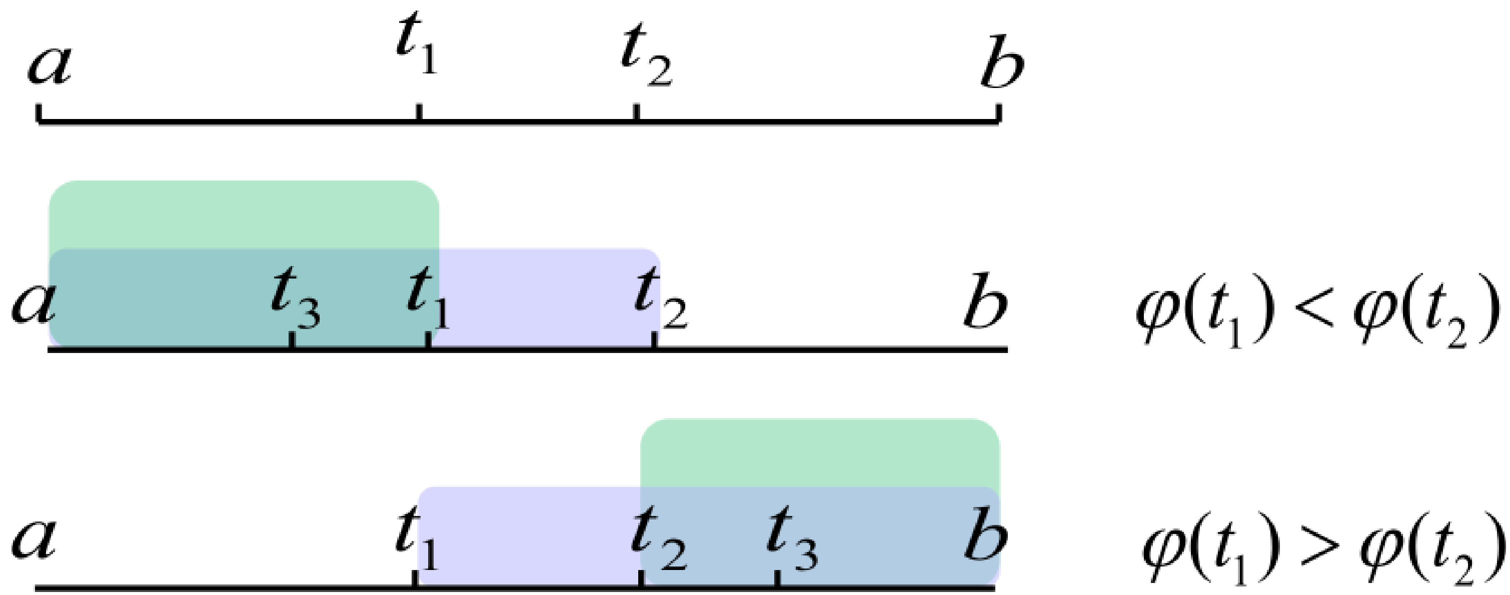
$\varphi(t_1) < \varphi(t_2)$

$\varphi(t_1) > \varphi(t_2)$

则有：
$$\frac{t_2 - a}{b - a} = \frac{b - t_1}{b - a} = c$$

一维搜索的0.618 法的原理

基本想法：每计算一个函数值能将区间压缩一个固定比值 c



为实现上述想法，必须：

$$\frac{t_2 - a}{b - a} = \frac{b - t_1}{b - a} = c \qquad \frac{t_1 - a}{t_2 - a} = \frac{b - t_2}{b - t_1} = c$$

$$\frac{t_2-a}{b-a}=\frac{b-t_1}{b-a}=c$$

$$\frac{t_1-a}{t_2-a}=\frac{b-t_2}{b-t_1}=c$$

$$\Rightarrow$$

$$t_1=b-c(b-a)$$

$$t_2=a+c(b-a)$$

$$t_1=a+c(t_2-a)$$

$$t_2=b-c(b-t_1)$$

由上面右边前三个方程可以解得

$$c^2=1-c$$

其中c的负数解被抛弃了

将前两个方程和新得到的方程代入第四个方程可使等式成立，所以满足上式的 c 即所求常数

解二次方程得正数解 $c=\frac{1}{2}\left(\sqrt{5}-1\right)\approx 0.618$

在单谷区间 $[a,b]$ 搜索局部最优解的0.618法

1）确定误差阈值 δ 及满足 $0.618^{n-1}(b-a)\le\delta$ 的 n

2）令 $a_0=a, b_0=b$

3）对于 $k=1,2,\cdots,n$ 依次完成以下运算

a）令 $t_k=a_{k-1}+0.618(b_{k-1}-a_{k-1})$ *（只需算一个）*

$t'_k=b_{k-1}-0.618(b_{k-1}-a_{k-1})$

b）计算 $\varphi(t_k)$ 和 $\varphi(t'_k)$ 中未知的数值

c）比较 $\varphi(t_k)$ 和 $\varphi(t'_k)$ 的大小确定 $[a_k,b_k]$

4）取所求局部最优解为 $0.5(a_n+b_n)$

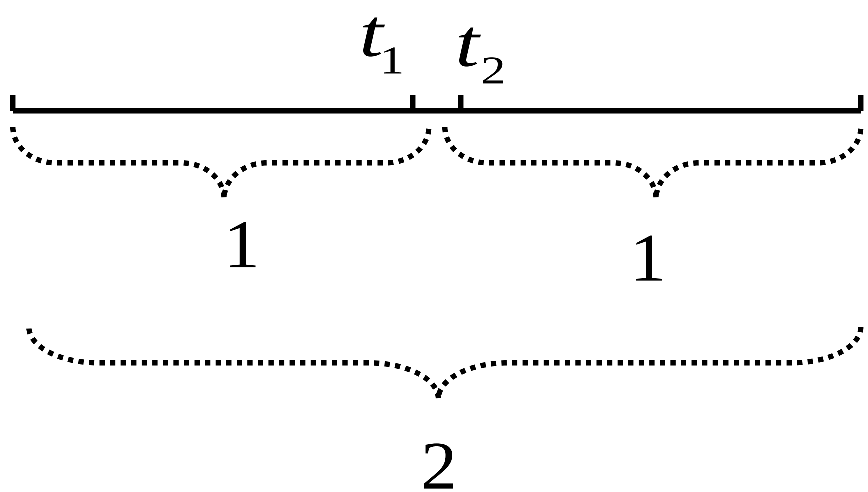
斐波那契（Fibonacci）法（最优选点方法）

用 F_n表示利用 n点函数值能将其压缩为一个单位长度的最大初始区间长度，即，通过计算区间中 n 点函数值，可以把长度为 F_n 的初始区间压缩为单位长度区间，但只要初始区间长度大于 F_n 就不能做到这一点

由于不计算函数值和只计算一点的函数值都不能压缩区间，所以

$$F_0 = F_1 = 1$$

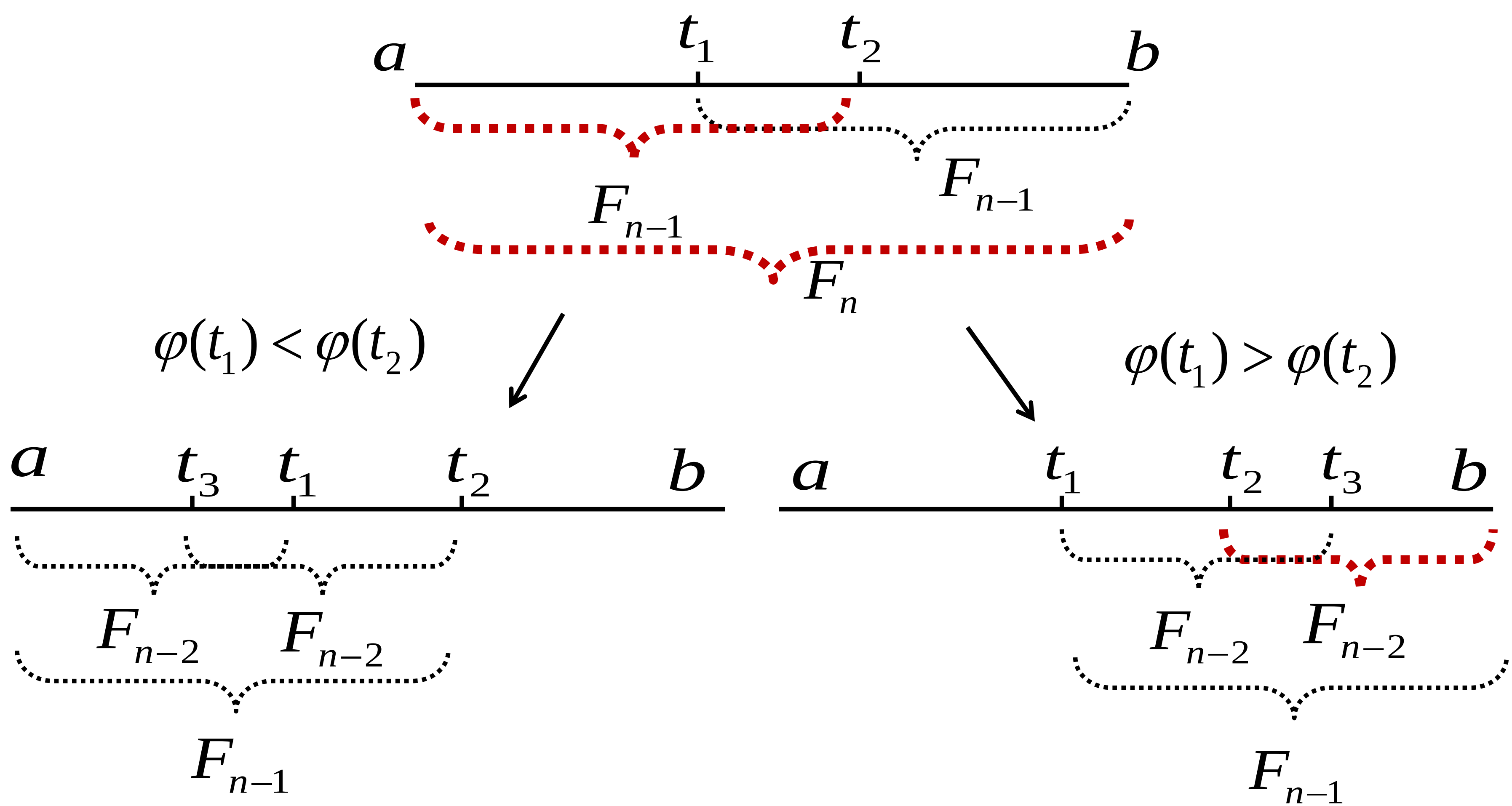
当 $n=2$ 时，由于两个分点可以任意接近中点，如下面的 t_1 和 t_2 所示，所以能将两个单位的区间压缩为一个单位的区间，即 $F_2=2$



前三个数的关系

$$F_0=1,\ F_1=1,\ F_2=F_1+F_0$$

一般情况下，F_n, F_{n-1}, F_{n-2} 必须满足下图所示关系



$$F_n = F_{n-1} + F_{n-2},$$

$$\frac{b-t_1}{b-a} = \frac{t_2-a}{b-a} = \frac{F_{n-1}}{F_n} \Longrightarrow t_1 = b + \frac{F_{n-1}}{F_n}(a-b),\ t_2 = a + \frac{F_{n-1}}{F_n}(b-a)$$

如何确定分点数 n？

假设初始区间是 $[a,b]$，容许误差是 $\delta>0$

将 δ 视为一个单位的长度，则在 $[a,b]$中一共有

$\dfrac{b-a}{\delta}$ 个单位长度，只要取 n 为满足

$$F_n \geq \frac{b-a}{\delta}$$

的**最小整数**，通过计算 n个分点的函数值，一定能将包含最优解的区间压缩为不大于一个单位的长度，即小于或等于 δ

在单谷区间 $[a,b]$ 搜索局部最优解的斐波那契法

1）确定误差阈值 δ 以及满足 $F_n \geq \delta^{-1}(b-a)$ 的 n

2）令 $a_0 = a, b_0 = b$

3）对于 $k = 1,2,\cdots,n-1$ 依次完成以下运算 *（只需算一个）*

a）令 $t_k = a_{k-1} + (F_{n-k}/F_{n-k+1})(b_{k-1} - a_{k-1})$

$$t_k' = b_{k-1} - (F_{n-k}/F_{n-k+1})(b_{k-1} - a_{k-1})$$

b）计算 $\varphi(t_k)$ 和 $\varphi(t_k')$ 中未知的数值

c）比较 $\varphi(t_k)$ 和 $\varphi(t_k')$ 的大小确定 $[a_k, b_k]$

4）取所求局部最优解为 $0.5(a_{n-1} + b_{n-1})$

0.618法和斐波那契法的关系

由 $F_n = F_{n-1} + F_{n-2}$ 可得 $\dfrac{F_n}{F_{n-1}} = 1 + \dfrac{F_{n-2}}{F_{n-1}}$

在上面右式令 $n \to \infty$ 并定义 $F_* = \lim\limits_{n\to\infty} \dfrac{F_{n-1}}{F_n}$

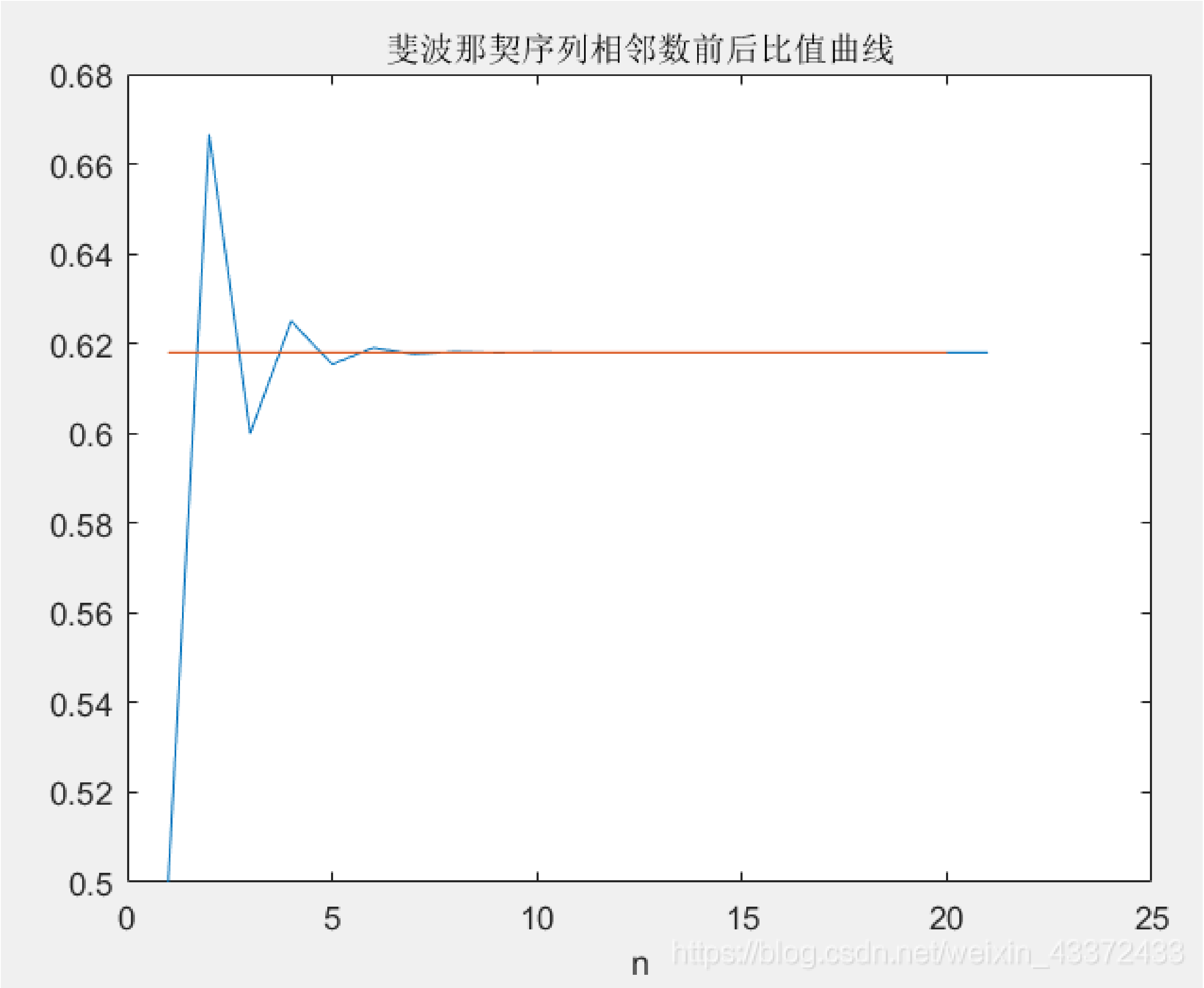
可得 $\dfrac{1}{F_*} = 1 + F_*$ 等价于 $(F_*)^2 + F_* - 1 = 0$

解二次方程得正数解 $F_* = \dfrac{1}{2}\left(\sqrt{5} - 1\right) \approx 0.618$

所以0.618法实际上就是用斐波那契法中分数数列的极限代替每个分数值所得到的方法

两个算法谁快?

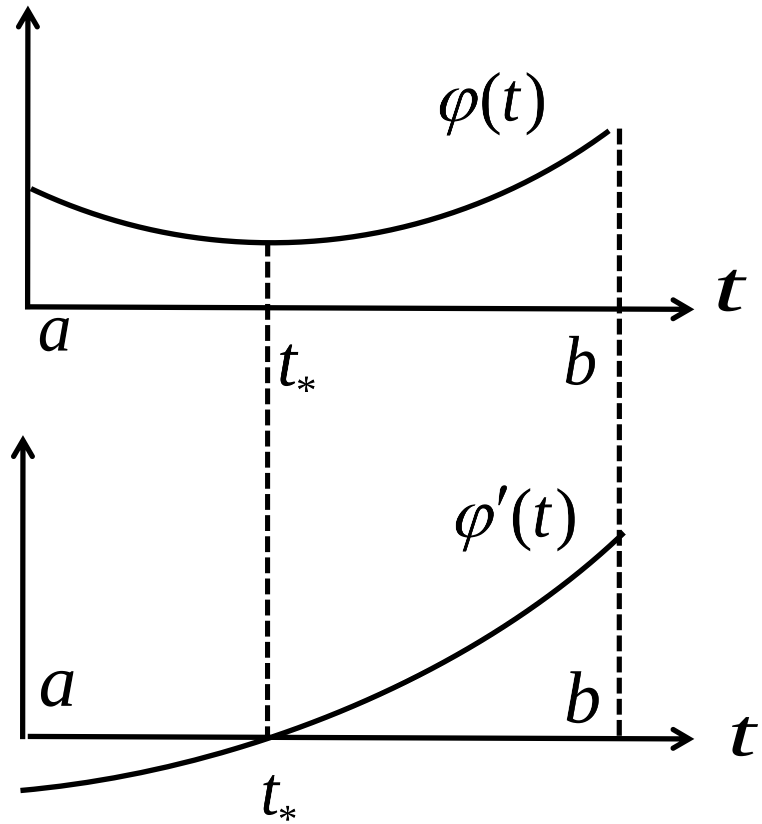
0.618法和斐波那契法的关系



利用一阶导数的精确搜索算法（区间对分法）

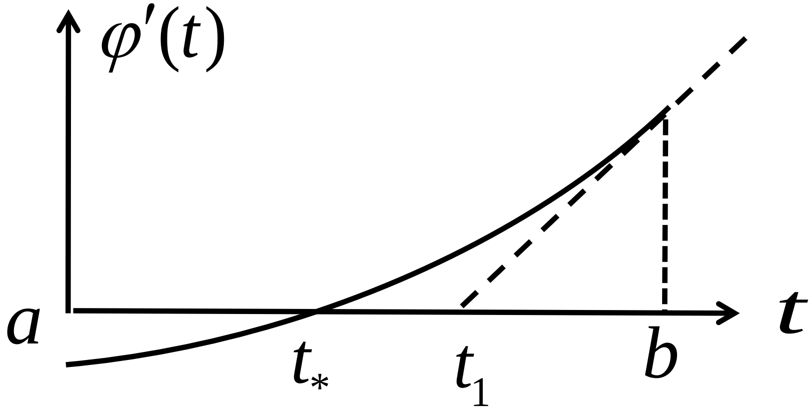
如右图，单谷区间 $[a,b]$ 一定满足 $\varphi'(a)<0,\varphi'(b)>0$

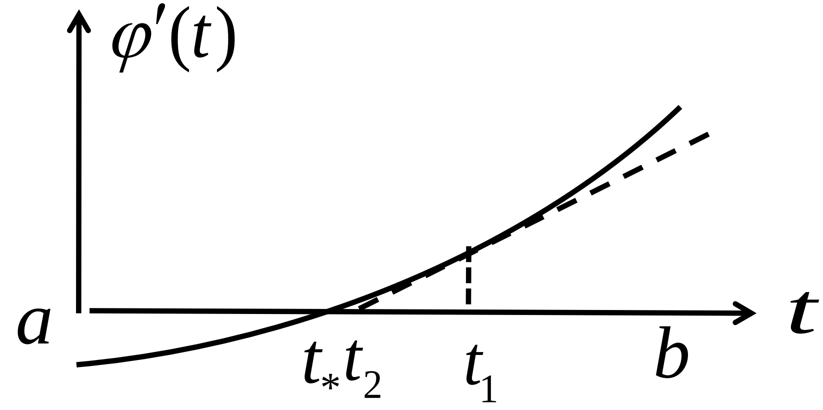
取 $t=0.5(a+b)$，若 $\varphi'(t)>0$，将区间压缩为 $[a,t]$，否则将区间压缩为 $[t,b]$



这种方法区间压缩比等于 0.5，比仅计算函数值的最好方法斐波那契法好，实际效果取决于导数计算量

利用二阶导数的精确搜索算法（Newton法）





如上左图，在 b 点用切线近似 $\varphi'(t)$ ，求该切线的零点

切线方程： $g(t) = \varphi'(b) + \varphi''(b)(t-b)$

$$g(t_1) = 0 \quad \Rightarrow \quad t_1 = b - \frac{\varphi'(b)}{\varphi''(b)}$$

再在 t_1 点重复上述过程 $\Rightarrow \quad t_2 = t_1 - \dfrac{\varphi'(t_1)}{\varphi''(t_1)}$

<u>如上所示</u>

<u>单谷区间</u>

<u>一定收敛</u>

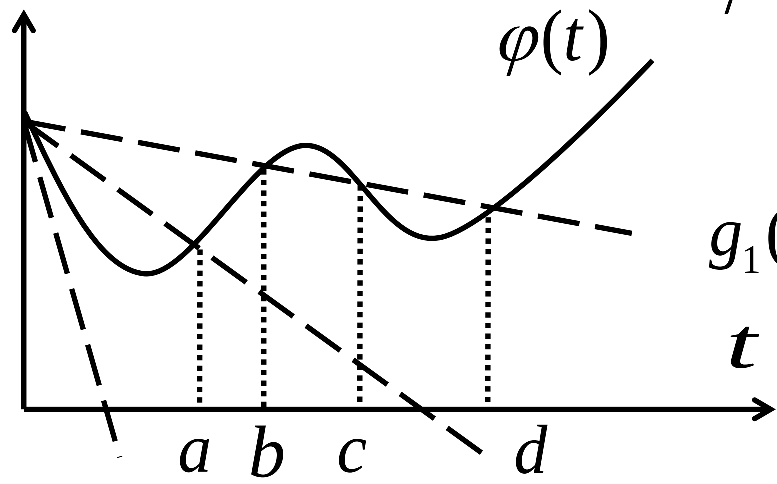
区间压缩算法　零阶算法

利用导数的精确搜索算法（区间对分法）　一阶算法

利用二阶导数的精确搜索算法（**Newton**法）二阶算法

非精确搜索的Goldstein法原理

$$\varphi'(0) < 0,\ 0 < m_1 < m_2 < 1$$



$$g_1(t) = \varphi(0) + m_1\varphi'(0)t$$

$$g_0(t) = \varphi(0) + \varphi'(0)t$$

$$g_2(t) = \varphi(0) + m_2\varphi'(0)t$$

基本想法： $\hat{t}$ 不能太小 $\Rightarrow$ $\varphi(\hat{t}) \geq g_2(\hat{t})$

函数减量不能太小 $\Rightarrow$ $\varphi(\hat{t}) \leq g_1(\hat{t})$

对于上图，合格的 $\hat{t}$ 属于区间 $[a,b]$ 或 $[c,d]$

非精确搜索的Goldstein法

1）选择 $\alpha > 1$，$0 < m_1 < m_2 < 1$，$0 = a_0 < t_0 < b_0 = \infty$，$k = 0$

2）计算 $\varphi(t_k)$，若 $\varphi(t_k) \le g_1(t_k)$，到 3，否则，令

$a_{k+1} = a_k$， $b_{k+1} = t_k$ ，然后到 5 （$\varphi(b_k) > g_1(b_k)$）

3）若 $\varphi(t_k) \ge g_2(t_k)$，停止迭代，输出 t_k ，否则，令

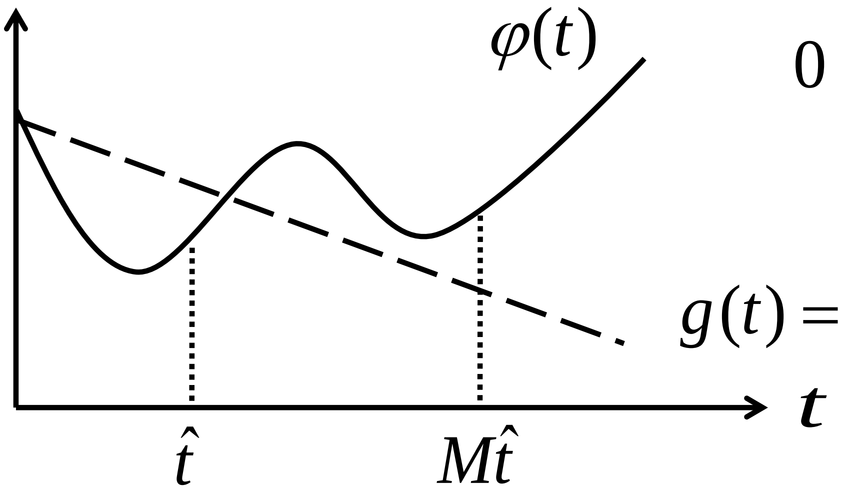
$a_{k+1} = t_k$ ， $b_{k+1} = b_k$ ，然后到 4 （$\varphi(a_k) < g_2(a_k)$）

4）若 $b_{k+1} < \infty$，到 5，否则，令 $t_{k+1} = \alpha t_k$ ，然后到 6

5）令 $t_{k+1} = 0.5(a_{k+1} + b_{k+1})$，然后到 6

6）用 $k+1$ 替换 k ，然后回到 2

Armijo法



$0<m<1<M$

减量不太小 $\Rightarrow \varphi(\hat{t})\le g(\hat{t})$　　　$\hat{t}$ 不太小 $\Rightarrow \varphi(M\hat{t})>g(M\hat{t})$

一种实现方法：选择足够大的 $\bar{t}>0$，取

$$\hat{t}=\max\left\{\beta(s)=\frac{\bar{t}}{2^{s-1}},\ s=1,2,\ldots \;\middle|\; \text{s.t. } \varphi\big(\beta(s)\big)\le\varphi(0)+\frac{1}{2}\beta(s)\varphi'(0)\right\}$$

这样的 $\hat{t}$ 对 $m=\dfrac{1}{2}$ 和 $M=2$ 满足上述条件

无约束优化

无约束优化问题

$$\min_{X \in R^n} f\left(X\right)$$

基本假定：目标函数具有二阶导数

凸优化问题的最优性条件

可导目标函数的凸问题最优解条件是：

充要条件

如有约束则D集合不是任意方向的集合

$$
\begin{aligned}
&x \text{ is optimal} \\
&\Leftrightarrow \text{if } x \text{ is optimal, for all } y \in D, \nabla f_0^T(x)(y-x) \geq 0
\end{aligned}
$$

证明如下：　充分条件

（ $\Leftarrow$ ）因为 $y \in D, \nabla f_0^T(x)(y-x) \geq 0, f_0(y) \geq f_0(x) + \nabla f_0^T(x)(y-x)$

所以 for all $y \in D, f_0(y) \geq f_0(x)$ ，即 x 是最优解

凸优化问题的最优性条件

必要条件

（ $\Rightarrow$ ）已知 x 是最优解，（下面是使用**反证法**进行证明）

假设条件：存在 $y \in D$ ，使得 $\nabla f_0^T(x)(y-x) < 0$

考虑函数 $z(t) = ty + (1-t)x,\ t \in [0,1]$ ，并且 $z(t) \in D$

则当 $t \to 0$ 时，利用假设条件可得

$$\frac{\mathrm{d} f_0(z(t))}{\mathrm{d} t} = \nabla f_0^T(x)(y-x) < 0$$

所以由**单调性**可得， $f_0(z(t)) < f_0(x)$ (其中 $z(0) = x$)，与已知**矛盾**

无约束凸优化问题 X^* 是严格局部最优解的充分条件：

$$\nabla f(X^*) = 0 \qquad \nabla^2 f(X^*) > 0$$

理由： $\nabla f(X^*) = 0 \quad \Rightarrow \quad \nabla^T f(X^*) D = 0, \quad \forall D \in R^n$

$$\Rightarrow \quad f(X^* + tD) - f(X^*) = \frac{1}{2} D^T \nabla^2 f(X^* + \xi D) D t^2, \quad \forall D \in R^n$$

$$\nabla^2 f(X^*) > 0 \quad \Rightarrow \quad \nabla^2 f(X^* + \xi D) > 0, \quad \forall \xi \in (0, \hat{\varepsilon})$$

$$\Rightarrow \quad f(X^* + tD) > f(X^*), \quad \forall D \in R^n, t \in (0, \hat{\varepsilon})$$

$$\Rightarrow \quad f(X) > f(X^*), \quad \forall X \in B(X^*, \varepsilon)$$

基本算法（下降方向法）：

1）任取 $X \in R^n$

2）如果在 X 处找不到下降方向，停止，否则，确定 X 处的下降方向 $D \in R^n$

3）直线搜索确定 t 满足 $f\left(X + t\,D\right) < f\left(X\right)$

4）用 $X + t\,D$ 替换 X，回到 2）继续迭代

实现算法的关键：如何确定下降方向 D ？

下降方向：负梯度方向

$$D = -\nabla f(X)$$

代入二阶泰勒展开

$$f(X+tD) = f(X) - \|\nabla f(X)\|^2 t + \frac{1}{2} D^T \nabla^2 f(X+\xi D) D t^2$$

$$\Rightarrow f(X+tD) - f(X)$$

$$= -t \left(\|\nabla f(X)\|^2 - \frac{1}{2} D^T \nabla^2 f(X+\xi D) D t \right)$$

只要 $\nabla f(X) \neq 0$，就有 $\|\nabla f(X)\| > 0$ ，一定存在 $\bar{t} > 0$

满足 $f(X+tD) < f(X), \forall 0 < t \leq \bar{t}$

所以负梯度方向是下降方向

梯度下降法

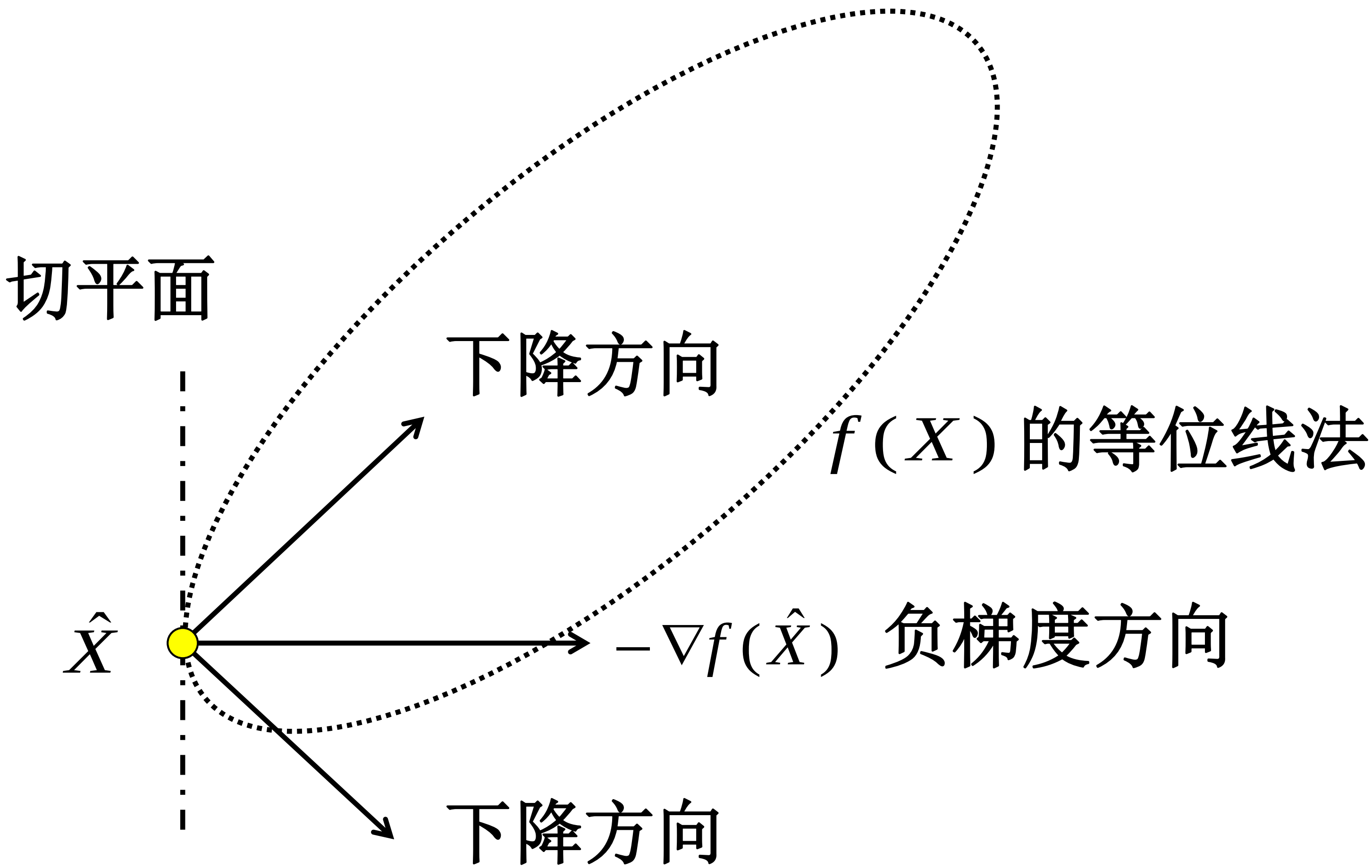
1）任取 $X \in R^n$，设定充分小的正数 ε

2）若 $\|\nabla f(X)\| \le \varepsilon$， 停止，否则令 $D = -\nabla f(X)$

3）直线搜索确定 t 满足 $f\left(X + tD\right) < f\left(X\right)$

4）用 $X + tD$ 替换 X，回到 2）继续迭代

负梯度方向的几何意义



<u>负梯度方向和切平面垂直</u>

精确搜索得到的点和搜索方向之间的关系

设 $\hat{X}'$是在 $\hat{X}$ 处沿下降方向 D 进行精确搜索所得到的点，即 $\hat{X}' = \hat{X} + \hat{t}D$ ，其中 $\hat{t}$ 是优化问题

$$\min_{t>0} f\left(\hat{X} + tD\right)$$

的最优解，应有 $$\frac{df(\hat{X} + \hat{t}D)}{dt} = 0$$

由 $$\frac{df(\hat{X} + tD)}{dt} = \frac{\partial f(\hat{X} + tD)}{\partial X^T} \frac{d(\hat{X} + tD)}{dt} = \nabla^T f(\hat{X} + tD)D$$

可得 $$\nabla^T f(\hat{X}')D = 0$$

即<u>所得到的点的梯度和所采用的搜索方向垂直</u>

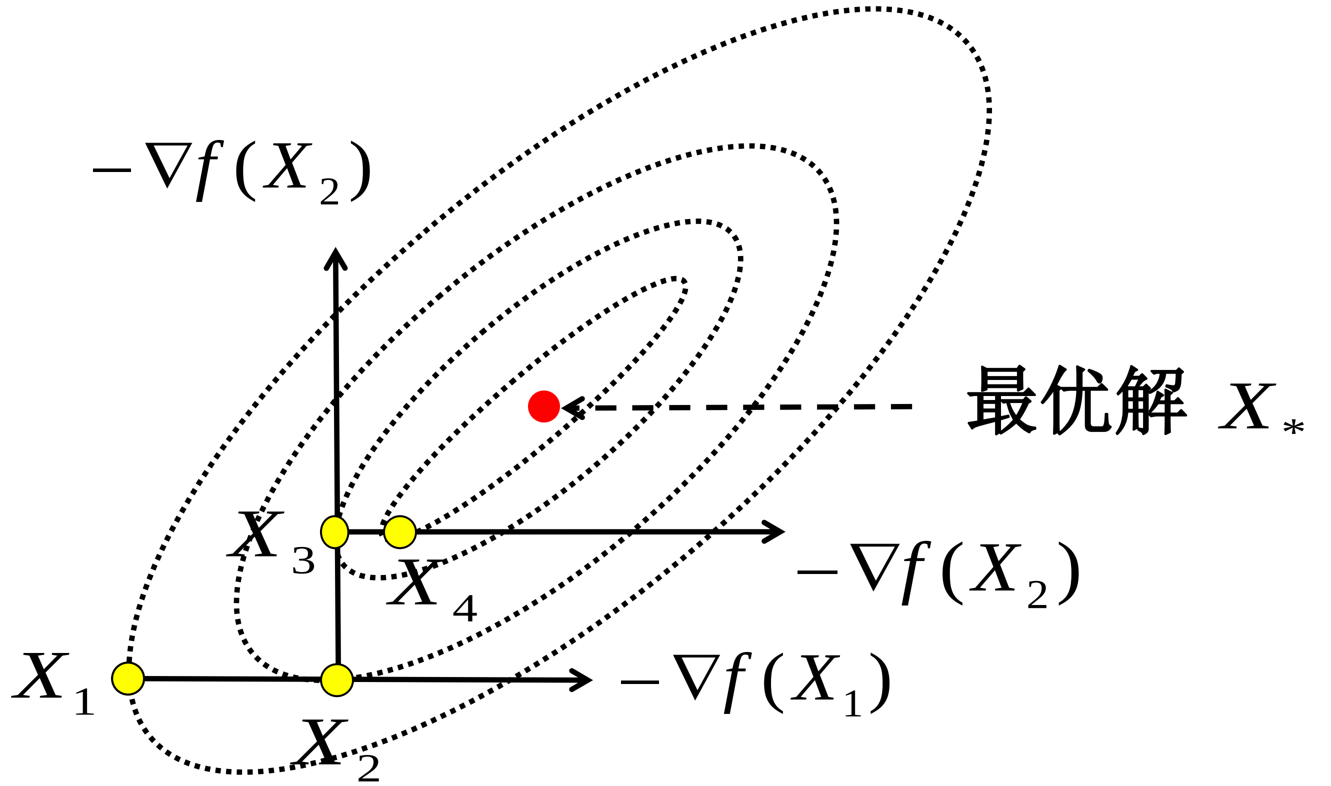
负梯度方向的特点

设 $\hat{X}'$是在 $\hat{X}$ 处沿负梯度方向 $D=-\nabla f(\hat{X})$ 进行一维搜索能得到的最好的点，由前面的结果可知

$$\nabla^T f(\hat{X}+\hat{t}D)\nabla f(\hat{X})=0$$

即沿负梯度方向精确搜索前进时，相邻两点的梯度互相垂直

负梯度方向的缺陷



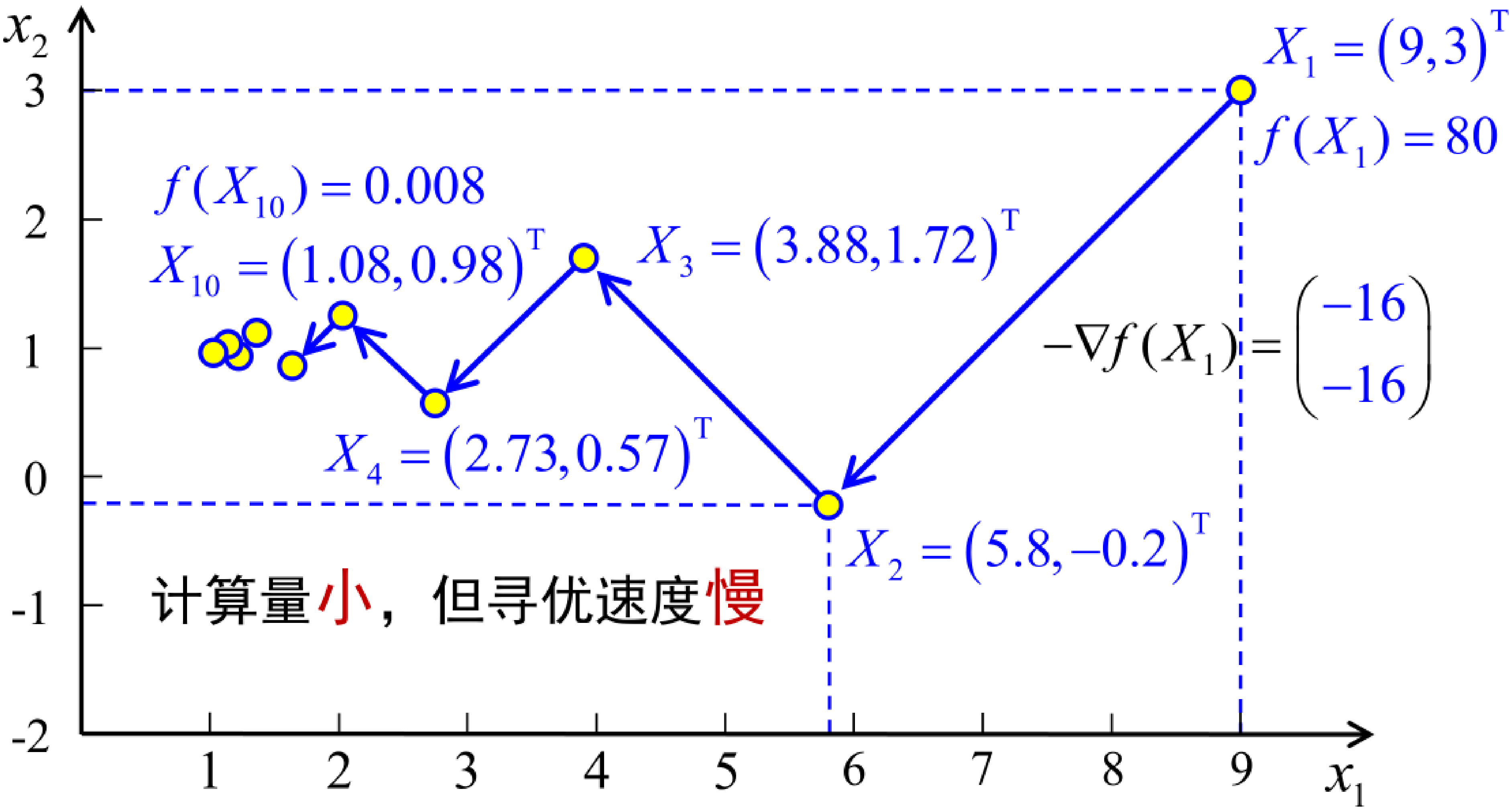
梯度下降法是沿锯齿状路线前进，接近最优解时一维搜索效率很低，前进速度很慢

梯度下降法的寻优过程

$$\min f(X) = x_1^2 + 4x_2^2 - 2x_1 - 8x_2 + 5$$

梯度下降法

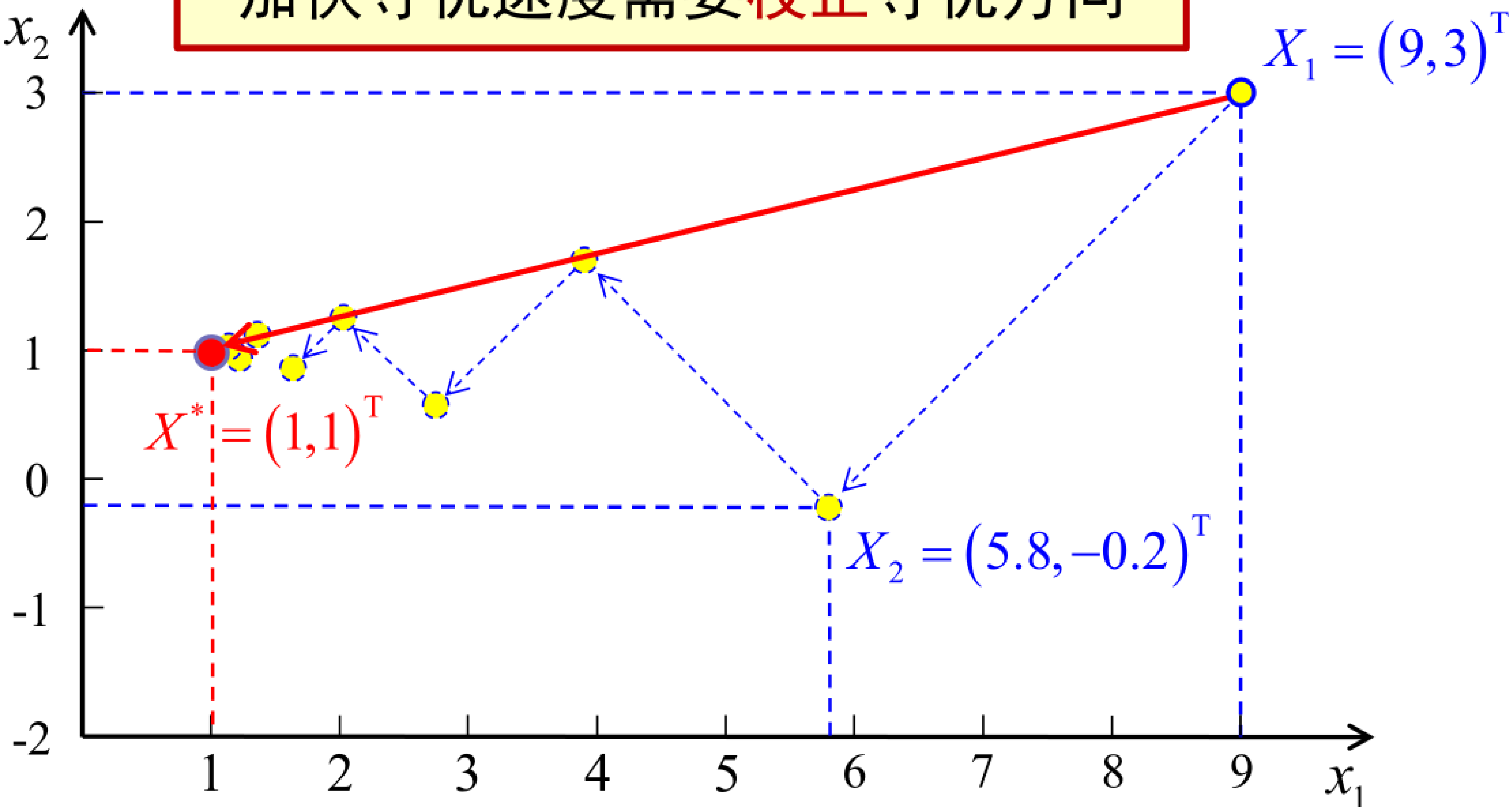
$$X_{k+1} = X_k - \lambda_k^* \nabla f(X_k)$$



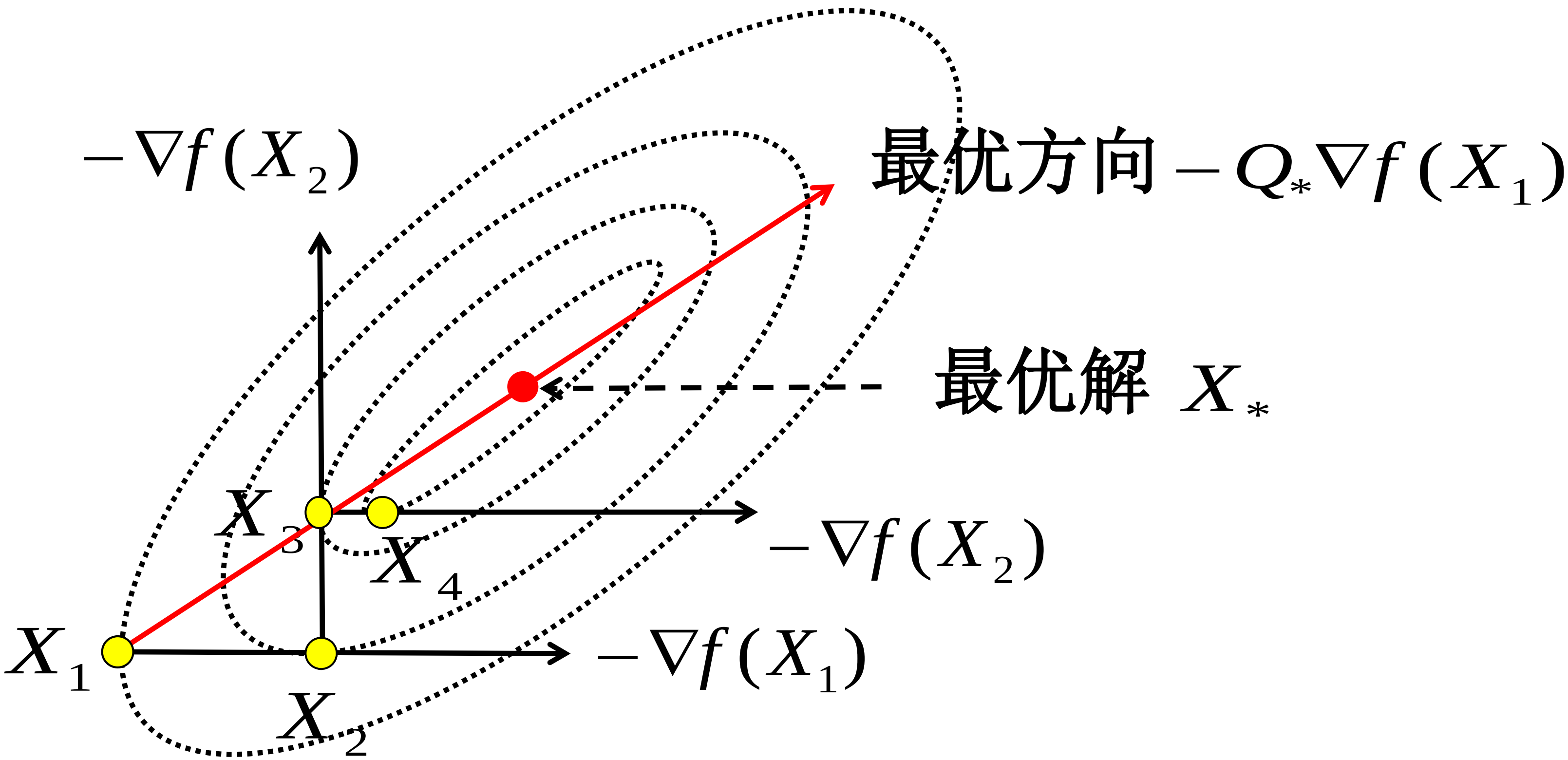
改进梯度下降法的思路

$$\min f(X) = x_1^2 + 4x_2^2 - 2x_1 - 8x_2 + 5$$

加快寻优速度需要校正寻优方向



克服负梯度方向缺陷的途径



用适当的正定矩阵（尺度矩阵）乘负梯度方向，其作用是对后者进行适当的旋转，以获得更好的方向

利用梯度方向生成其它下降方向

任取一个正定矩阵 $Q \in R^{n \times n}$，令 $D = -Q\nabla f(X)$

代入二阶泰勒展开可得

$$f(X+tD) - f(X)$$

$$= -t\left(\nabla^T f(X) Q \nabla f(X) - \frac{1}{2} D^T \nabla^2 f(X + \xi D) D t \right)$$

只要 $\nabla f(X) \neq 0$，就有 $\nabla^T f(X) Q \nabla f(X) > 0$，一定存在 $\bar{t} > 0$ 满足

$$f(X+tD) < f(X), \forall 0 < t \leq \bar{t}$$

所以 D 是下降方向

给定方向的二阶泰勒展开

$$f(X+tD)=f(X)+\nabla^T f(X)Dt+\frac{1}{2}D^T\nabla^2 f(X+\xi D)Dt^2$$

$\Rightarrow$ <u>下降方向的充分条件 $\nabla^T f(X)D<0$</u>

<u>下降方向的必要条件 $\nabla^T f(X)D\le 0$</u>

<u>**最速下降方向：**</u>

$$\min\left\{\nabla^T f(X)D \,\middle|\, \text{s.t. } \|D\|=1\right\}$$

$$\Leftrightarrow \max\left\{-\nabla^T f(X)D \,\middle|\, \text{s.t. } \|D\|=1\right\} \Rightarrow \hat{D}$$

ℓ_1 范数 $\|D\|_1 = \sum_{i=1}^{n} |d_i|$

$$\hat{d}_i = \begin{cases} \operatorname{sgn}\left(-\dfrac{\partial f(X)}{\partial x_i}\right) & \text{if } \left|\dfrac{\partial f(X)}{\partial x_i}\right| = \|\nabla f(X)\|_\infty \\ 0 & \text{if } \left|\dfrac{\partial f(X)}{\partial x_i}\right| \neq \|\nabla f(X)\|_\infty \end{cases}$$

$$\nabla f(X)^T \hat{D} = -\|\nabla f(X)\|_\infty$$

$$-\nabla^T f(X) D \le \sum_{i=1}^{n} \left|\frac{\partial f(X)}{\partial x_i}\right| |d_i| \le \max_{1\le i\le n} \left|\frac{\partial f(X)}{\partial x_i}\right| \sum_{i=1}^{n} |d_i|$$

$$\Downarrow$$

$$\|\nabla f(X)\|_\infty$$

ℓ_p 范数 $\|D\|_p = \left(\sum_i |d_i|^p \right)^{\frac{1}{p}}$， $p > 1$

$$\hat{d}_i = \mathrm{sgn}\left(-\frac{\partial f(X)}{\partial x_i} \right) \left| \frac{\partial f(X)}{\partial x_i} \right|^{q-1} \left(\|\nabla f(X)\|_q \right)^{-\frac{q}{p}}, \forall i$$

$$\nabla f(X)^T \hat{D} = -\|\nabla f(X)\|_q \qquad \left(\frac{1}{q} = 1 - \frac{1}{p} \right)$$

ℓ_∞ 范数 $\|D\|_\infty = \max_{1 \le i \le n} |d_i|$

$$\hat{d}_i = \mathrm{sgn}\left(-\frac{\partial f(X)}{\partial x_i} \right), \forall i$$

$$\nabla f(X)^T \hat{D} = -\|\nabla f(X)\|_1$$

负梯度方向（ℓ_2范数最速下降方向）

$$\hat{D} = -\nabla f(X)\left(\left\|\nabla f(X)\right\|_2\right)^{-1} \Leftrightarrow -\nabla f(X)$$

$$\nabla f(X)^T \hat{D} = -\left\|\nabla f(X)\right\|_2$$

牛顿方向（$\left\|D\right\|_{\nabla^2 f(X)} = \left(D^T \nabla^2 f(X) D\right)^{\frac{1}{2}}$的最速下降方向）

$$\hat{D} = \frac{-\left(\nabla^2 f(X)\right)^{-1} \nabla f(X)}{\left(\nabla f(X)^T \left(\nabla^2 f(X)\right)^{-1} \nabla f(X)\right)^{\frac{1}{2}}} \Leftrightarrow -\left(\nabla^2 f(X)\right)^{-1} \nabla f(X)$$

$$\nabla f(X)^T \hat{D} = -\nabla f(X)^T \left(\nabla^2 f(X)\right)^{-1} \nabla f(X)$$

正定二次函数的最优方向

$$f(X)=\frac{1}{2}X^TAX+B^TX+C$$

$$\nabla f(X)=AX+B=0 \quad \Rightarrow \quad X_*=-A^{-1}B$$

$$\nabla^2 f(\hat{X})=A$$

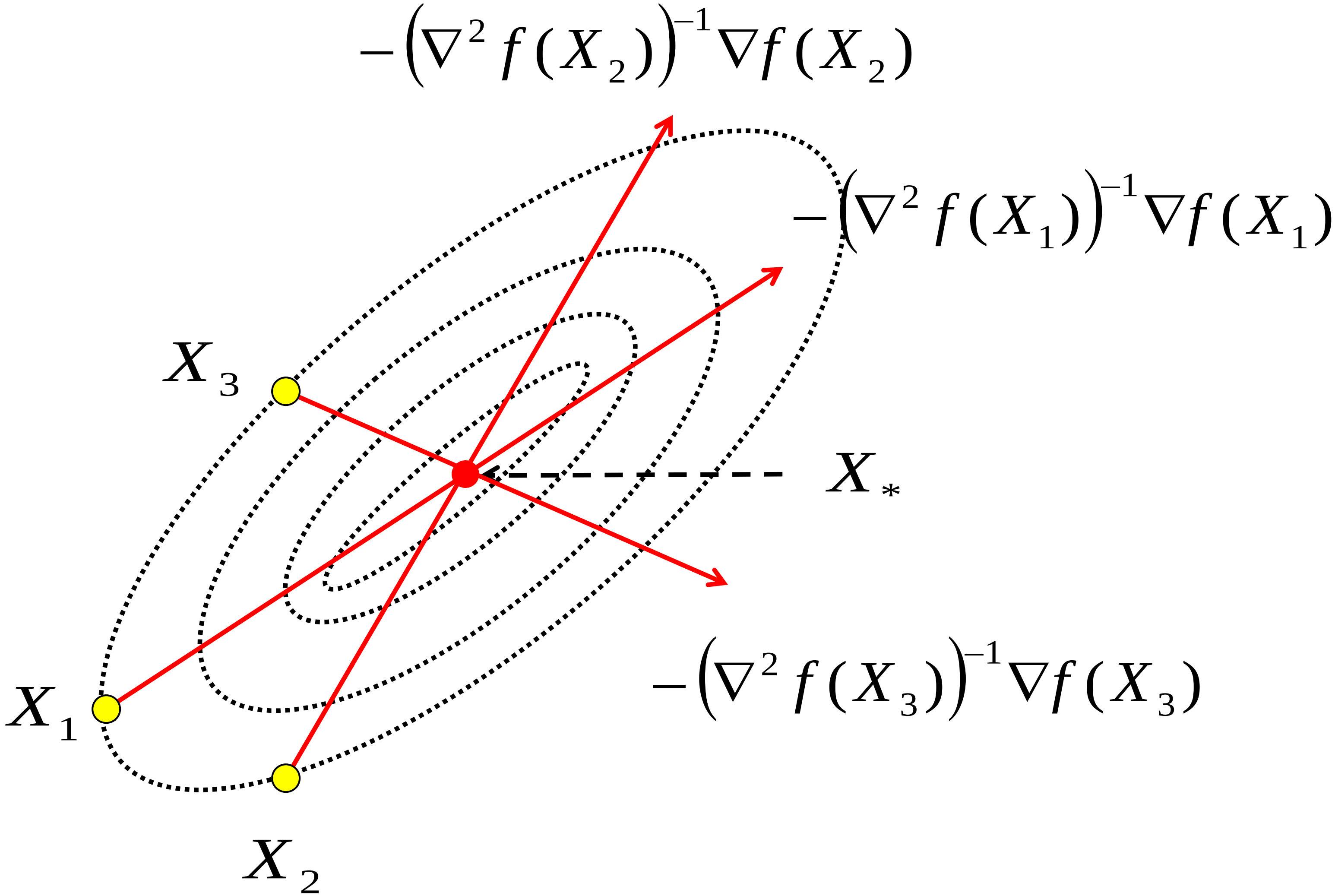
令 $$D_*=X_*-X=-A^{-1}\left(AX+B\right)=-\left(\nabla^2 f(X)\right)^{-1}\nabla f(X)$$

$$t_*=1$$

$\Rightarrow$ $$\min_{t>0} f(X+tD_*)=f(X+t_*D_*)=f(X_*)$$

$\Rightarrow$ 最优方向： $D=-\left(\nabla^2 f(\hat{X})\right)^{-1}\nabla f(\hat{X})$

正定二次函数的牛顿方向



阻尼牛顿法（广义牛顿法）

1）任取 $X \in R^n$，设定充分小的正数 ε

2）若 $\|\nabla f(X)\| \le \varepsilon$，停止，否则令 $D = -\left(\nabla^2 f(X)\right)^{-1} \nabla f(X)$

3）直线搜索确定 t 满足 $f\left(X + tD\right) < f\left(X\right)$

4）用 $X + tD$ 替换 X，回到 **2**）继续迭代

阻尼牛顿法的缺陷

1）每步迭代要计算 $\left(\nabla^2 f(X)\right)^{-1}$ ，计算量大

2）$\left(\nabla^2 f(X)\right)^{-1}$ 可能不存在（对于严格凸问题不会）

3）$\left(\nabla^2 f(X)\right)^{-1}$ 可能不正定，$D = -\left(\nabla^2 f(X)\right)^{-1}\nabla f(X)$ 不是下降方向（对于严格凸问题不会）

用阻尼牛顿法求解下列问题

$$\min f(x) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

初始点 $x^{(1)} = (0,0)^T$。

在初始点的梯度和Hessian矩阵分别为

$$\nabla f(x) = \begin{bmatrix} 4x_1^3 + x_2 \\ x_1 + 2(1 + x_2) \end{bmatrix} \quad \nabla^2 f(x) = \begin{bmatrix} 12x_1^2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\nabla f(x^{(1)}) = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \nabla^2 f(x^{(1)}) = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

在初始点的牛顿方向为

$$d^{(1)} = -\nabla^2 f(x^{(1)})^{-1} \nabla f(x^{(1)}) = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

在初始点沿牛顿方向进行一维精确搜索

$$\min \varphi(t) = f\left(x^{(1)} + td^{(1)}\right)$$

$$= 16t^4 + 1$$

可以得到

$$\varphi'(t) = 64t^3 = 0$$

$$t^{(1)} = 0$$

显然，用阻尼牛顿法不能产生新的点，而初始点并不是无约束优化问题的极小点。

牛顿方向失效的原因在于初始点的Hessian矩阵非正定！

严格凸优化问题

梯度下降，其它各种改进型梯度下降（包括牛顿方向），都能收敛到全局最优解

非凸优化问题

梯度下降能收敛到局部最优解，其它方法不一定收敛