大学数学实验

实验7 无约束优化

清 华 大 学 数 学 科 学 系

优化模型和算法的重要意义

最优化：在一定条件下，寻求使目标最大(小)的决策

最优化是工程技术、经济管理、科学研究、社会生活中经常遇到的问题, 如:

结构设计　资源分配　生产计划　运输方案

解决决策问题的手段

- **经验积累，主观判断**
- **作试验，比优劣**
- **建立数学模型，求解最优策略**

(最)优化理论是运筹学的基础

OR/ MS/ DS	
	运筹学(OR: Operations/Operational Research)
	管理科学(MS: Management Science)
	决策科学 (DS: Decision Science)

优化(Optimization), 规划(Programming)

无约束优化　线性规划　非线性规划　整数规划　组合优化　不确定规划　多目标规划　目标规划　网络优化　动态规划

数学规划的一般形式

$$\min_x \ z = f(x),\ x \in \Omega \subseteq R^n$$

x～决策变量，f～目标函数，Ω～可行域

$$\min_x \quad z = f(x) \qquad (1)$$

$$s.t. \quad g_i(x) \le 0,\ i = 1,2,\cdots m \qquad (2)$$

- **可行解（只满足(2)）与最优解（满足(1),(2)）**
- **无约束优化（只有(1)）与约束优化（(1),(2)）**
- **实际问题一般总有约束，何时可用无约束优化处理?**

无约束优化的主要内容

1. **优化问题的最优解条件；算法模式**
2. **无约束优化的基本方法:**

   梯度法，牛顿法，拟牛顿法
3. **线性、非线性最小二乘法**
4. **优化工具箱的使用**
5. **实际问题中的无约束优化模型**

实例1 产销量安排

某厂生产两个牌号的同一种产品，如何确定产量使利润最大

假设A　产销平衡

牌号	产量	成本	价格
甲	x1	q1	p1
乙	x2	q2	p2

假设B　p随x (两种牌号)增加而减小，呈线性关系

$$p_1 = b_1 - a_{11}x_1 - a_{12}x_2,\ b_1, a_{11}, a_{12} > 0,\ a_{11} > a_{12}$$

$$p_2 = b_2 - a_{21}x_1 - a_{22}x_2,\ b_2, a_{21}, a_{22} > 0,\ a_{22} > a_{21}$$

实例1 产销量安排

假设C q随x (本牌号)增加而减小，呈负指数关系

$$q_1 = r_1 e^{-\lambda_1 x_1} + c_1,\quad r_1,\lambda_1,c_1 > 0$$

$$q_2 = r_2 e^{-\lambda_2 x_2} + c_2,\quad r_2,\lambda_2,c_2 > 0$$

目标 **利润最大**

$$\max_{x_1,x_2} z(x_1,x_2) = (p_1 - q_1)x_1 + (p_2 - q_2)x_2$$

实例2 饮酒驾车血液中的酒精含量（续）

已建模型 由实验1：酒精含量$c(t)$随时间t的变化规律

$$c(t) = c_0 e^{-kt} \qquad y = \ln c(t),\quad a_1 = -k,\quad a_2 = \ln c_0$$

$$\Rightarrow \qquad y = a_1 t + a_2$$

实际数据

t	1	2	3	4	5	6	7	8	9	10	11	12
$c(t)$	82	77	68	51	41	38	35	28	25	18	15	12

$$y_i = a_1 t_i + a_2,\quad i = 1,2,\cdots,n.$$

方程个数超过了未知数个数，称为超定方程组

在普通意义下一般无解，需要在新的准则下定义它的解

超定方程组与曲线拟合

由已知数据计算已知函数关系中的系数，

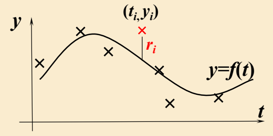
称为数据拟合（或曲线拟合, curve fitting）

数据(曲线)拟合问题的一般提法

已知一组数据（t_i,y_i), i=1,…,n，寻求一个函数（曲线）$y=f(t)$，使$f(t)$在某种准则下与所有数据点最为接近，即曲线拟合得最好。

当函数/曲线的形式确定后，待定系数 ➔ 超定方程组

曲线拟合与最小二乘准则



使点(t_i,y_i) 与曲线 $y=f(t)$ 的距离r_i尽量小,i=1,…,n;

记 $r=(r_1,r_2,\ldots,r_n)$

使$f(t_i)$与y_i之差的平方和(即图中r_i的平方和)最小，称为最小二乘(Least Square) 准则 （等价于r的2-范数最小）

对实例2： $\min\limits_{a_1,a_2} \sum\limits_{i=1}^{12} |a_1 t_i + a_2 - y_i|^2$

其他准则：最小化r_i的绝对值的和（r的1-范数）

最小化r_i中的最大者（r的∞-范数）

线性最小二乘拟合的基本思路

先选定一组基函数$\varphi_0(t)$, $\varphi_1(t)$, … $\varphi_m(t)$, $m+1<n$, 令

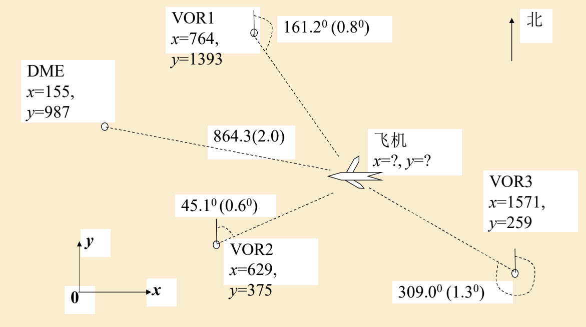
$f(t)=\beta_0\varphi_0(t)+\beta_1\varphi_1(t)+\ldots+\beta_m\varphi_m(t)$

$$f(t_i) = y_i \;(i=1,\cdots,n) \Leftrightarrow \begin{cases} \beta_0\varphi_0(t_1) + \beta_1\varphi_1(t_1) + \cdots + \beta_m\varphi_m(t_1) = y_1 \\ \cdots\cdots \\ \beta_0\varphi_0(t_n) + \beta_1\varphi_1(t_n) + \cdots + \beta_m\varphi_m(t_n) = y_n \end{cases}$$

$$\Phi = \begin{bmatrix} \varphi_0(t_1) & \varphi_1(t_1) & \cdots & \varphi_m(t_1) \\ \cdots\cdots \\ \varphi_0(t_n) & \varphi_1(t_n) & \cdots & \varphi_m(t_n) \end{bmatrix} \qquad \beta = (\beta_0,\beta_1,\cdots,\beta_m)^T \quad y = (y_1,\cdots y_n)^T$$

$$\Rightarrow \Phi\beta = y \Rightarrow \min_{\beta} \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} |f(t_i) - y_i|^2 = \|\Phi\beta - y\|_2^2$$

实例3 飞机精确定位问题



飞机与监控台（图中坐标和测量距离的单位是“公里”）

飞机精确定位模型

已知数据：设备位置坐标分别为$(x_i, y_i), i=1,...4$;

记测量角度为θ_i，角度误差为$\sigma_i, i=1,...,3$;

记测量距离为d_4，距离误差为σ_4.

要求计算：飞机位置坐标（x, y）

不考虑误差因素　　量纲不符！

$$\mathrm{atan2}(x-x_i, y-y_i)=\theta_i$$
$$\sqrt{(x-x_4)^2+(y-y_4)^2}=d_4$$

超定方程组，非线性最小二乘！

$$\min_{x,y}\sum_{i=1}^{3}\left(\frac{\mathrm{atan2}(x-x_i, y-y_i)-\theta_i}{\theta_i}\right)^2+\left(\frac{\sqrt{(x-x_4)^2+(y-y_4)^2}-d_4}{d_4}\right)^2$$

飞机精确定位模型

考虑误差因素　　误差非均匀分布！

$$\theta_i-\sigma_i \le \mathrm{atan2}(x-x_i, y-y_i) \le \theta_i+\sigma_i$$
$$d_4-\sigma_4 \le \sqrt{(x-x_4)^2+(y-y_4)^2} \le d_4+\sigma_4$$

非线性规划

$\mathrm{Min}\ x;\quad \mathrm{Min}\ y;\quad \mathrm{Max}\ x;\quad \mathrm{Max}\ y.$

有人也可能会采用其他目标，如：

以距离为约束，优化角度误差之和（或平方和）；

或以角度为约束，优化距离误差.

仅部分考虑误差!

角度与距离的“地位”不应不同！

飞机精确定位模型

误差一般服从什么分布？

正态分布！

不同的量纲如何处理？

归一化处理！

$\mathrm{Min}\ \ E(x,y)=$　　无约束非线性最小二乘模型

$$\sum_{i=1}^{3}\left(\frac{\mathrm{atan2}(x-x_i, y-y_i)-\theta_i}{\sigma_i}\right)^2+\left(\frac{d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}}{\sigma_4}\right)^2$$

角度需要进行预处理，

如利用**atan2**函数，值域**(-pi, pi)**

多选题　2分　　设置

根据本教材的观点，数学规划模型的三要素是：

A 模型是最大化问题还是最小化问题

B 目标函数

C 决策变量

D 模型的参数

E 约束条件

F 模型是连续优化还是离散优化

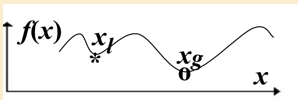
提交

无约束优化:最优解的分类和条件

给定一个函数 $f(x)$，寻找 x^* 使得 $f(x^*)$最小，即

$$\min_x f(x) \quad \textbf{其中}\quad x=(x_1, x_2, \cdots, x_n)^T \in \Re^n$$

局部最优解　　**全局最优解**



⇩

必要条件　$\nabla f(x^*)=(f_{x_1}, \cdots, f_{x_n})^T=0$

充分条件　$\nabla f(x^*)=0, \nabla^2 f(x^*)>0$　$\nabla^2 f=\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]_{n\times n}$

Hessian阵

多选题　2分　　设置

关于无约束极小化问题的解法，下列说法正确的是

A 一般采用直接法求解

B 一般采用迭代法求解

C 迭代法一般先确定搜索步长，再确定搜索方向

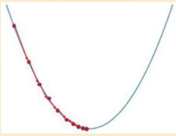
D 迭代法一般先确定搜索方向，再确定搜索步长

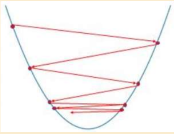
E 最速下降法是收敛最快的迭代法

提交

求解无约束优化的基本思路

在 $\Re^n$ 中某一点，确定一个搜索方向及沿该方向步长➔使目标函数下降的新点





迭代步骤

Step 1 初始化：初始点x^0，终止准则等

Step 2 迭代改进：方向d^k，步长α^k

$$x^{k+1} = x^k + \alpha^k d^k$$

$f(x^{k+1}) < f(x^k)$ （下降法，下山法）

Step 3 终止检验：得到近优解或$k+1 \Rightarrow k$转2

选择d^k,α^k 使 f 下降更快 $\Rightarrow$ 不同算法

搜索方向的选择

1 最速下降法（梯度法）

暂不考虑搜索步长，可设$\alpha^k=1$

将 $f(x^{k+1})$ 在 x^k 点作泰勒展开，只保留一阶项，有

$$f(x^{k+1}) = f(x^k + d^k) = f(x^k) + \nabla f^T(x^k) d^k$$

下降方向 $\nabla f^T(x^k) d^k < 0$

最速下降方向 $d^k = -\nabla f(x^k)$ （负梯度方向）

迭代改进格式 $x^{k+1} = x^k - \nabla f(x^k)$

算法特点 **初始阶段改进较快，最优解附近改进较慢**

（“最速”：局部性质）　（“锯齿现象”）

2 Newton方法

将$f(x^{k+1})$在x^k点作泰勒展开至二阶项，用d替代d^k

$$f(x^{k+1}) = f(x^k + d) = f(x^k) + \nabla f^T(x^k) d + \frac{1}{2} d^T \nabla^2 f(x^k) d$$

求d使$f(x^{k+1})$极小$\Rightarrow$右端对d导数为0 $\Rightarrow \nabla f(x^k) + \nabla^2 f(x^k) d = 0$

牛顿方程 $\nabla^2 f(x^k) d = -\nabla f(x^k)$

牛顿方向 $d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k)$

迭代格式 $x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$

特点 局部2阶收敛;需计算Hessian阵,它可能病态或不正定

比较 解 $F(x)=0$ 的牛顿法 $x^{k+1} = x^k - [F'(x^k)]^{-1} F(x^k)$

$F(x) \Leftrightarrow \nabla f(x)$

3 拟Newton方法

目的 **不计算Hessian阵，克服病态、不正定、计算复杂等缺陷，同时保持收敛较快的优点**

思路 **回顾解方程组 F(x)=0的拟牛顿法**

$$x^{k+1} = x^k - [F'(x^k)]^{-1} F(x^k) \Rightarrow x^{k+1} = x^k - (A^k)^{-1} F(x^k)$$

使A^k满足 $A^k(x^k - x^{k-1}) = F(x^k) - F(x^{k-1})$

用迭代方法$A^k = A^{k-1} + \Delta A^{k-1}$ 计算A^k

优化问题 $\underset{x}{Min}\ f(x)$　$\nabla f(x)$相当 $F(x)$

$\nabla^2 f(x)$相当$F'(x)$, $\nabla^2 f$不一定正定，构造正定阵G代替$\nabla^2 f$

3 拟Newton方法(续)

设在第k步, G^k已得到, $H^k=(G^k)^{-1}$,可计算

$$x^{k+1} = x^k - H^k \nabla f(x^k)$$

记 $\Delta x^k = x^{k+1} - x^k$, $\Delta f^k = \nabla f(x^{k+1}) - \nabla f(x^k)$

G^{k+1}满足**拟牛顿条件：（相应的方向称拟牛顿方向）**

$$G^{k+1} \Delta x^k = \Delta f^k \text{ 或 } \Delta x^k = H^{k+1} \Delta f^k$$

构造 迭代公式$G^{k+1} = G^k + \Delta G^k$ 或$H^{k+1} = H^k + \Delta H^k$

于是有 $x^{k+2} = x^{k+1} - H^{k+1} \nabla f(x^{k+1})$

3 拟Newton方法(续)

3.1 Davidon-Fletcher-Powell (DFP) 公式

$$\Delta H^k = \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta f^k} - \frac{H^k \Delta f^k (\Delta f^k)^T H^k}{(\Delta f^k)^T H^k \Delta f^k}$$

$$\Delta G^k = (1 + \frac{(\Delta x^k)^T G^k \Delta x^k}{(\Delta x^k)^T \Delta f^k}) \frac{\Delta f^k (\Delta f^k)^T}{(\Delta x^k)^T \Delta f^k} - \frac{\Delta f^k (\Delta x^k)^T G^k + G^k \Delta x^k (\Delta f^k)^T}{(\Delta x^k)^T \Delta f^k}$$

3.2 Broyden-Fletcher-Goldfarb-Shanno (BFGS) 公式

$$\Delta G^k = \frac{\Delta f^k (\Delta f^k)^T}{(\Delta f^k)^T \Delta x^k} - \frac{G^k \Delta x^k (\Delta x^k)^T G^k}{(\Delta x^k)^T G^k \Delta x^k}$$

$$\Delta H^k = (1 + \frac{(\Delta f^k)^T H^k \Delta f^k}{(\Delta x^k)^T \Delta f^k}) \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta f^k} - \frac{\Delta x^k (\Delta f^k)^T H^k + H^k \Delta f^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta f^k}$$

多选题 2分 设置

关于无约束极小化问题的解法，下列说法正确的是

- A 最速下降法就是梯度(下降)法
- B 最速下降法是收敛速度最快的算法
- C 最速下降法是牛顿法的特例
- D 拟牛顿法是牛顿法的特例
- E 以上都不对

提交

搜索步长的确定

一维搜索 (线搜索，Line Search)确定步长的方法

问题 给定x^k和方向d^k, 确定步长$\alpha^k>0$ 使得

$$\min_{\alpha} f(x^k+\alpha d^k) \sim \text{一维优化问题}$$

优化算法 $\phi(\alpha)=f(x^k+\alpha d^k)$

(精确线搜索) $\phi(\alpha^k)=\min\phi(\alpha)$

$$\frac{df}{d\alpha}\Big|_{\alpha^k}=0 \Rightarrow (d^k)^T\nabla f(x^k+\alpha^k d^k)=0$$

非精确线搜索： α^k**使函数值下降一个可接受的量即可**

(例如：Wolfe 准则；Armijo 准则）----略去

精确线搜索(Line Search) $\min_{\alpha} f(x^k+\alpha d^k)$

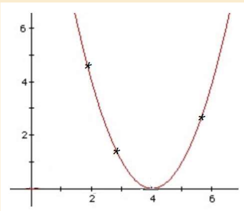
黄金分割(0.618)法、Fibonacci法、Newton切线法、割线法、2次（抛物线）或 3次多项式插值法等

- **通常先确定搜索区间[a, b]** **(bracketing phase)**

  [例如] 进退法（下页）

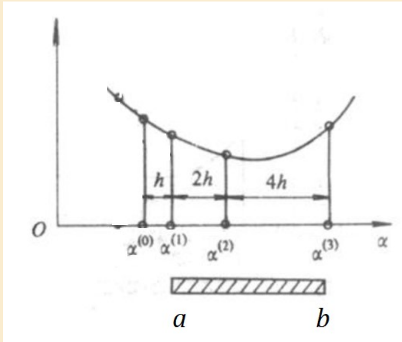
  ➔找到“高-低-高”三个点

- **然后精细搜索** **(sectioning phase)**



单峰(unimodal)函数；单谷函数

搜索区间：进退法（示意）



搜索区间：进退法

步1 初值: 选取$\alpha_0\ge 0$, $h_0>0$. 计算$\phi_0:=\phi(\alpha_0)$. 置$k:=0$.

步2 令$\alpha_{k+1}=\alpha_k+h_k$, 计算$\phi_{k+1}=\phi(\alpha_{k+1})$. 若$\phi_{k+1}\le\phi_k$, 转步3, 否则转步4.

步3 加大步长. 令$h_{k+1}=\tau h_k$, $\alpha=\alpha_k$, $\alpha_k=\alpha_{k+1}$, $\phi_k=\phi_{k+1}$, $k=k+1$, 转步2. （τ常取2）

步4 反向搜索或输出.
若$k=0$, 令$h_1=-h_0$, $\alpha=\alpha_1$, $\alpha_1=\alpha_0$, $\phi_1=\phi_0$, $k=1$, 转步2;
否则停止迭代, 令 $a=\min\{\alpha,\alpha_{k+1}\}$, $b=\max\{\alpha,\alpha_{k+1}\}$.

投票 最多可选1项 设置

已知单变量连续函数f(x)在（a, b）区间端点取值异号，求f(x)=0在（a, b）区间上的根，0.618法与二分法相比，你觉得哪个更好？

- A 0.618法
- B 二分法
- C 不分伯仲
- D 我不知道，不能草率下结论

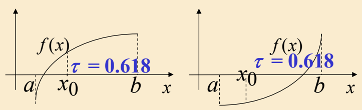
提交

大学数学实验

非线性方程的基本解法

0.618法 （黄金分割法？）

$f(a)<0$
$f(b)>0$



假设概率各为1/2(不太合理)，则区间长度缩短

期望值： $\tau * 1/2 + (1-\tau) * 1/2 = 1/2$

假设零点均匀分布，则区间长度缩短

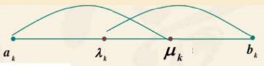
期望值： τ^2 + (1- τ) ^2 >= 1/2 (二分法更好)

大学数学实验

线搜索的黄金分割法(0.618法)

针对单峰（单谷）函数

$\tau = 0.618$ $\tau = 0.618$



$b_k - \lambda_k = \mu_k - a_k,$

缩短率

$b_{k+1} - a_{k+1} = \tau_k (b_k - a_k), \frac{1}{2} < \tau_k < 1 (1)$

$\mu_k - a_k = \tau_k (b_k - a_k)$ → $\mu_k = a_k + \tau_k (b_k - a_k)$ (2)

$b_k - \lambda_k = \tau_k (b_k - a_k)$ → $\lambda_k = a_k + (1-\tau_k)(b_k - a_k)$ (3)

且下次迭代时，$[a_{k+1}, b_{k+1}]$区间内旧的点仍可利用！

大学数学实验

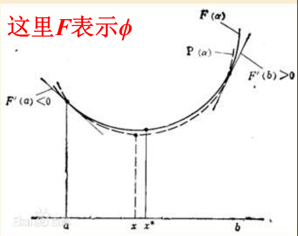
三次插值法（Matlab中使用）

Davidon (1959) 提出

先找区间$[a,b]$: $a < b$,
$F'(a) < 0$, $F'(b) > 0$

用 a, b 两点的函数值、导数值，得插值 $P(\alpha)$
(三次Hermite插值)

这里F表示ϕ



$$P(\alpha) = a_1 \left(\alpha - \alpha^{(0)}\right)^3 + a_2 \left(\alpha - \alpha^{(0)}\right)^2 + a_3 \left(\alpha - \alpha^{(0)}\right) + a_4, \quad (1)$$

通常取 $\alpha^{(0)} = a$, 则 $a_4 = F(a)$, $a_3 = F'(a)$

大学数学实验

三次插值法（Matlab中使用）

$$P(\alpha) = a_1 \left(\alpha - \alpha^{(0)}\right)^3 + a_2 \left(\alpha - \alpha^{(0)}\right)^2 + a_3 \left(\alpha - \alpha^{(0)}\right) + a_4, \quad (1)$$

$a_4 = F(a)$, $a_3 = F'(a)$ $F(b) = P(b)$, $F'(b) = P'(b)$

$$a_1 = \frac{(b-a)[F'(b)+F'(a)] + 2[F(a)-F(b)]}{(b-a)^3}$$

$$a_2 = \frac{3[F(b)-F(a)] - (b-a)[F'(b)+2F'(a)]}{(b-a)^2}$$

求解 $P'(\alpha) = 0$ (两根)，并要求 $P''(\alpha) > 0$, 直接得到

$$\alpha^* = \alpha^{(0)} - \frac{a_3}{a_2 + \sqrt{a_2^2 - 3a_1 a_3}}$$

当 $a_1 = 0$：退化成书上的二次(抛物线)插值

多选题 2分 设置

关于无约束极小化问题的解法，下列说法正确的是

- A 一般采用直接法求解
- B 一般采用迭代法求解
- C 迭代法一般先确定搜索步长，再确定搜索方向
- D 迭代法一般先确定搜索方向，再确定搜索步长
- E 最速下降法是收敛最快的迭代法

提交

大学数学实验

超定方程组与曲线拟合

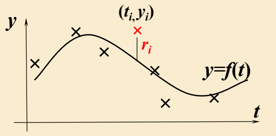
由已知数据计算已知函数关系中的系数，
称为数据拟合（或曲线拟合, curve fitting）

数据(曲线)拟合问题的一般提法

已知一组数据 (t_i, y_i), $i=1,\dots,n$, 寻求一个函数（曲线）$y=f(t)$, 使$f(t)$在某种准则下与所有数据点最为接近，即曲线拟合得最好。

当函数/曲线的形式确定后，待定系数 → 超定方程组

曲线拟合与最小二乘准则



使点 (t_i,y_i) 与曲线 $y=f(t)$ 的距离r_i尽量小,$i=1,\ldots,n$;

记 $r=(r_1,r_2,\ldots,r_n)$

使$f(t_i)$与y_i之差的平方和(即图中r_i的平方和)最小，称为最小二乘(Least Square) 准则（等价于r的2-范数最小）

对实例 2： $\min\limits_{a_1,a_2}\sum_{i=1}^{12}|a_1t_i+a_2-y_i|^2$

其他准则：最小化r_i的绝对值的和（r的1-范数）

最小化r_i中的最大者（r的∞-范数）

线性最小二乘拟合的基本思路

先选定一组基函数$\varphi_0(t),\varphi_1(t),\ldots\varphi_m(t)$, $m+1<n$, 令

$f(t)=\beta_0\varphi_0(t)+\beta_1\varphi_1(t)+\ldots+\beta_m\varphi_m(t)$

$$f(t_i)=y_i\quad(i=1,\cdots,n)\iff\begin{cases}\beta_0\varphi_0(t_1)+\beta_1\varphi_1(t_1)+\cdots+\beta_m\varphi_m(t_1)=y_1\\ \cdots\cdots\\ \beta_0\varphi_0(t_n)+\beta_1\varphi_1(t_n)+\cdots+\beta_m\varphi_m(t_n)=y_n\end{cases}$$

$$\Phi=\begin{bmatrix}\varphi_0(t_1)&\varphi_1(t_1)&\cdots&\varphi_m(t_1)\\ \cdots\cdots\\ \varphi_0(t_n)&\varphi_1(t_n)&\cdots&\varphi_m(t_n)\end{bmatrix}\qquad \beta=(\beta_0,\beta_1,\cdots,\beta_m)^T\quad y=(y_1,\cdots y_n)^T$$

$$\Rightarrow\ \Phi\beta=y\ \Rightarrow\ \min_\beta\sum_{i=1}^n r_i^2=\sum_{i=1}^n|f(t_i)-y_i|^2=\|\Phi\beta-y\|_2^2$$

线性最小二乘法: 求解

$$J(\beta_0,\beta_1,\cdots,\beta_m)=\|\Phi\beta-y\|_2^2=\sum_{i=1}^n[f(t_i)-y_i]^2=\sum_{i=1}^n[\sum_{k=0}^m\beta_k\varphi_k(t_i)-y_i]^2$$

$$\frac{\partial J}{\partial\beta_k}=0\quad(k=1,\cdots m)\ \Rightarrow\ \begin{cases}\sum_{i=1}^n\varphi_0(t_i)[\sum_{k=0}^m\beta_k\varphi_k(t_i)-y_i]=0\\ \sum_{i=1}^n\varphi_1(t_i)[\sum_{k=0}^m\beta_k\varphi_k(t_i)-y_i]=0\\ \cdots\cdots\\ \sum_{i=1}^n\varphi_m(t_i)[\sum_{k=0}^m\beta_k\varphi_k(t_i)-y_i]=0\end{cases}$$

$$\Phi=\begin{bmatrix}\varphi_0(t_1)&\varphi_1(t_1)&\cdots&\varphi_m(t_1)\\ \cdots\cdots\\ \varphi_0(t_n)&\varphi_1(t_n)&\cdots&\varphi_m(t_n)\end{bmatrix}$$

$$\Phi^T\Phi\beta=\Phi^Ty\ \Rightarrow\ \beta=(\Phi^T\Phi)^{-1}\Phi^Ty$$

线性最小二乘法：思考

$$\Phi=\begin{bmatrix}\varphi_0(t_1)&\varphi_1(t_1)&\cdots&\varphi_m(t_1)\\ \cdots\cdots\\ \varphi_0(t_n)&\varphi_1(t_n)&\cdots&\varphi_m(t_n)\end{bmatrix}\qquad \Phi^T\Phi\beta=\Phi^Ty\qquad \beta=(\Phi^T\Phi)^{-1}\Phi^Ty$$

问题 1　**怎样选择 $\{\varphi_0(t),\ldots,\varphi_m(t)\}$，以保证系数 $\{\beta_0,\ldots,\beta_m\}$有唯一解?**

分析　**$\{\beta_0,\ldots,\beta_m\}$有唯一解 ← $\Phi^T\Phi$可逆 ← Rank($\Phi^T\Phi$)=m**

← Rank(Φ)=m　← Φ列满秩

← $\{\varphi_0(t),\ldots,\varphi_m(t)\}$在$t_i$点线性无关 ($i$=1, …$n$)

问题 2　**为什么要规定$m+1<n$?**

分析　强令$f(t_i)=\beta_0\varphi_0(t_i)+\ldots+\beta_m\varphi_m(t_i)=y_i$ (i=1, …n)

(即曲线 $f(t)$过全部数据点，此时J=0） 得

$$\Phi\beta=y,\quad \Phi=\begin{bmatrix}\varphi_0(t_1)&\cdots&\varphi_m(t_1)\\ \cdots\cdots\\ \varphi_0(t_n)&\cdots&\varphi_m(t_n)\end{bmatrix}\quad \beta=[\beta_0,\cdots,\beta_m]^T\quad y=[y_1,\cdots,y_n]^T$$

若$m+1>n$，β一般有无穷多解

若$m+1=n$，Φ可逆时β有唯一解

若$m+1<n$，超定方程组, β一般无解；求最小二乘解

问题 3　**线性最小二乘中的“线性”指的是什么?**

$f(t)$ 对β线性，于是求解线性方程组　$(\Phi^T\Phi)\beta=\Phi^Ty$

非线性最小二乘拟合

问题　**给定(t_i,y_i), $i=1,\ldots,n$, 拟合一个函数$y=f(x,t)$，**

其中x为待定的参数向量，f对x非线性。

记误差　$r_i(x)=f(x,t_i)-y_i$　$r(x)=(r_1(x),\cdots r_n(x))^T$

优化模型　$\min\limits_x R(x)=\frac{1}{2}r^T(x)r(x)$

根据目标函数是 $r(x)$ 的二次函数的特点

构造简单算法

注：非线性最小二乘拟合也可用来求非线性超定方程组的最小二乘解（只需要将方程组看成这里的$r(x)$=0）

非线性最小二乘拟合

$R(x)=\frac{1}{2}r^T(x)r(x)$

记 $r(x)$ 的雅各比阵为 $J(x)=(\partial r_i/\partial x_j)_{n\times m}$

$\nabla R=J(x)^T r(x) \qquad \nabla^2 R=J(x)^T J(x)+S$

$S=\sum_{i=1}^{n} r_i(x)\nabla^2 r_i(x) \qquad \nabla^2 r_i(x)=(\partial^2 r_i/\partial x_k \partial x_l)_{m\times m}$

讨论 **·牛顿法要计算Hessian矩阵，其中S计算量大**

·若 f 对 x 线性, 则化为线性最小二乘拟合,此时S=0

特定算法考虑如何忽略或近似矩阵S。

非线性最小二乘拟合

Gauss-Newton算法：忽略矩阵S

$\nabla R=J(x)^T r(x) \qquad \nabla^2 R=J(x)^T J(x)$

牛顿方程 $\nabla^2 f(x^k)d=-\nabla f(x^k)$

f用R代替，下降方向d^k满足

$J(x^k)^T J(x^k)d^k=-J(x^k)r(x^k)$

G-N算法 **收敛性依赖 f 对 x 的线性程度，及偏差r的大小**

非线性最小二乘拟合

Levenbery-Marquardt算法：G-N算法修正

$J(x^k)^T J(x^k)d^k=-J(x^k)r(x^k)$

⇩ 防止 J^TJ 出现病态

$(J(x^k)^T J(x^k)+\alpha^k I)d^k=-J(x^k)r(x^k)$

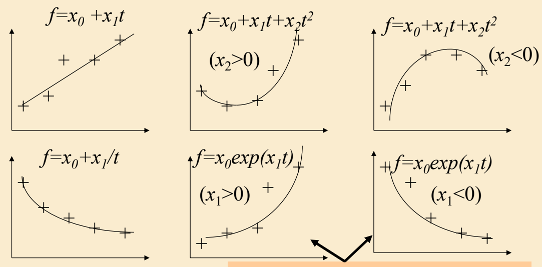
其中α^k>0为修正参数.

L-M算法 **d^k位于牛顿方向(α^k很小)和负梯度方向(α^k很大)之间**

拟合函数形式的选取

1. 通过机理分析建立数学模型来确定 $f(t)$: $f(x, t)$或$f(\beta,t)$

2. 将数据 (t_i,y_i) i=1, …n 作图，通过直观判断确定 $f(t)$



f 对 a 非线性，但可线性化

多选题 2分 设置

给定(t_i, y_i), i=1,…,n, 用最小二乘法拟合一个函数 y=$f(x, t)$，其中x为待定的参数向量。下列说法正确的是：

A 最小二乘法的目标是使n个点上的误差的绝对值之和最小

B 最小二乘法的目标是使n个点上的误差的平方和最小

C 当 f 关于 x 是线性函数时称为线性最小二乘法

D 当 f 关于 t 是线性函数时称为线性最小二乘法

提交

用MATLAB解无约束优化

模型: $\underset{x}{Min}\ f(x),\ x\in R^n$

f.m ~ f(x)的m文件名

x0 ~初始点； x ~最优解

基本用法：

```
x=fminbnd(@f,lb,ub)
x=fminunc(@f,x0)
x=fminunc(@f,x0,options)
x=fminsearch(@f,x0,options)
```

例1: $\min\ (3\sin x+x)$

$x\in[1,8]$

Exam0701.m

例2: $\min \dfrac{x^2}{a}+\dfrac{y^2}{b}$

其中 $a=b=2$

Exam0702.m

用MATLAB解无约束优化

- `fminbnd` ← 'golden section search, parabolic interpolation'
- `fminsearch` ← 'Nelder-Mead simplex direct search'
- `fminunc` ← 'quasi-newton' (default) or 'trust-region'
  - 'bfgs', the default
  - 'dfp'
  - 'steepdesc'(梯度法)

非线性最小二乘法

$$\min_{lb\le x\le ub} R(x)=r^T(x)r(x) \qquad r_i(x)=f(x,t_i)-y_i \qquad r(x)=(r_1(x),\cdots r_n(x))^T$$

```
[x,resnorm,res,exitf,out,lambda,jacob]=
lsqcurvefit(@f,x0,t,y,lb,ub,options)
[x,resnorm,res,exitf,out,lambda,jacob]=
lsqnonlin(@fun,x0,lb,ub,options)
```

输入的用法与fminunc类似，但注意:

f.m~f(x)的m文件名: function y=f(x,t)

fun.m~r(x)的m文件名: function r=fun(x,t,y)

输出 resnom=r(x)^T*r(x)，res=r(x)（误差向量）

多选题 2分

设置

关于Matlab的优化求解器，以下说法正确的是

- A fminbnd 适合于最多10个决策变量的问题
- B fminsearch 适合于目标函数不连续的问题
- C fminsearch 可以求解有多个决策变量的问题
- D fminunc 优于 fminbnd 和 fminsearch
- E 以上都不对

提交

非线性最小二乘法

$$\min_{lb\le x\le ub} R(x)=r^T(x)r(x)$$

例3　用下面一组数据拟合系数r, k : $c(t)=re^{-kt}$

t	0.25	0.5	1	1.5	2	3	4	6	8
c	19.21	18.15	15.36	14.10	12.89	9.32	7.45	5.24	3.01

exam0703fit.m　　exam0703lsq.m

实例1　产销量安排

原问题

$$p_i=b_i-a_{i1}x_1-a_{i2}x_2,\ b_i,a_{i1},a_{i2}>0,\ i=1,2,$$

$$q_i=r_ie^{-\lambda_ix_i}+c_i,\ r_i,\lambda_i,c_i>0,\ i=1,2,$$

$$\max_{x_1,x_2} z(x_1,x_2)=(p_1-q_1)x_1+(p_2-q_2)x_2$$

已知数据

$$\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}=\begin{pmatrix}1&0.1\\0.2&2\end{pmatrix}\quad b=\begin{pmatrix}100\\280\end{pmatrix}$$

$$r^T=(30\quad 100),\quad \lambda^T=(0.015\quad 0.02),\quad c^T=(20\quad 30)$$

$$\min\quad f(x)=-(b_1-a_{11}x_1-a_{12}x_2-r_1e^{-\lambda_1x_1}-c_1)x_1-(b_2-a_{21}x_1-a_{22}x_2-r_2e^{-\lambda_2x_2}-c_2)x_2$$

实例1　产销量安排

初始点的选择　忽略成本及价格中的 a_{12}, a_{21}　$\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}=\begin{pmatrix}1&0.1\\0.2&2\end{pmatrix}$

问题简化为　$\min\quad f(x)=-(b_1-a_{11}x_1)x_1-(b_2-a_{22}x_2)x_2$

其解为$x_1=b_1/2a_{11}=50$，　$x_2=b_2/2a_{22}=70$

作为原问题的初始点

命令和最优解　x=fminunc(@f, x0)　　shili0701.m

x = 23.9025　　62.4977　　y = 6.4135e+003

即甲产量为23.9025，乙产量为62.4977，最大利润为6413.5

实例2 饮酒驾车血液中的酒精含量

$y = a_1 t + a_2 \quad \varphi_0 = t, \varphi_1 = 1 \quad \Phi\beta = y$ 无约束线性最小二乘模型

方法 1：套用法方程，求解方程组

$$\Phi^T \Phi \beta = \Phi^T y \qquad \beta = (\Phi^T \Phi)^{-1} \Phi^T y$$

方法 2：左除

$$\beta = \Phi \backslash y$$

方法 3：多项式拟合(polyfit命令)

m=1; a = polyfit(t, y, m); % 降幂排列

shili0702.m

输出结果：a = - 0.1747 4.6723
k = 0.1747 c0 = 106.9412

实例3 飞机精确定位模型

$Min \ E(x,y) =$ 无约束非线性最小二乘模型

$$\sum_{i=1}^{3}\left(\frac{\operatorname{atan2}(x-x_i, y-y_i)-\theta_i}{\sigma_i}\right)^2 + \left(\frac{d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}}{\sigma_4}\right)^2$$

角度需要进行预处理，
如利用atan2函数, 值域(-pi, pi)

shili0703.m

飞机坐标(978.31，723.98), 误差平方和0.6685 (<< 4)

MATLAB优化工具箱及更多的功能

- 优化工具箱主要功能
  - 无约束优化
  - 约束优化
  - ……
- 控制参数的设置
  - 精度控制
  - 算法选择
  - 梯度与雅可比矩阵的计算（数值微分 / 解析表达式）
  - 初始点的选取
  - ……

控制精度，观察中间结果，控制迭代次数等

控制参数设定: optimoptions （旧optimset不推荐）

```
Optimset                //显示控制参数
optimset optfun         //显示optfun的控制参数
opt=optimset            //控制参数设为[](即缺省值)
opt=optimset(optfun)//optfun控制参数设缺省值
Opt=optimset('par1',val1,'par2',val2,...)
Opt=optimset(oldopts,'par1',val1,...)
opt=optimset(oldopts,newopts)
val=optimget(opt,'par1','par2',…)
val=optimget(opt,'par1','par2',…, default)
```

主要控制参数(对大规模/中等规模算法均有效)

```
Diagnostics  'on' | {'off'} //是否显示诊断信息
Display      'off' | 'iter' |
            'final' | 'notify' //显示信息的级别
GradObj      'on' | {'off'}//是否采用分析梯度
Jacobian     'on' | {'off'}
             //采用分析Jacob阵 (用于约束优化中)
LargeScale   'on' | {'off'}//是否采用大规模算法
MaxFunEvals  最大函数调用次数
MaxIter      最大迭代次数
TolCon       约束的控制精度 (用于约束优化中)
TolFun       函数值的控制精度
TolX         解的控制精度
```

更多输出：最优值等

最一般的输出形式
[x,f,exitflag,out,grad,hess]=fminunc(...)

```
f           目标函数值
exitflag    >0收敛,0达到函数或迭代次数, <0不收敛
Output
  iterations      实际迭代次数
  funcCount       实际函数调用次数
  algorithm       实际采用的算法
  cgiterations    实际PCG迭代次数 (大规模算法用)
  stepsize        最后迭代步长 (中等规模算法用)
  firstorderopt   一阶最优条件 (梯度的范数)
grad  目标函数的梯度
hess  目标函数的Hessian矩阵
```

例 4 $\min \frac{x_1^2}{a}+\frac{x_2^2}{b}$, $a=10, b=1$

Exam0704.m

用optimset控制参数选择

算法选择：optimoptions中参数控制

Algorithm:'trust-region' | 'quasi-newton'

(当前的）缺省值为“quasi-newton”

(“trust-region”必须要求用户提供梯度函数)

拟牛顿法搜索方向的选择

- HessUpdate = ‘bfgs’ （BFGS，缺省值）
- HessUpdate = ‘dfp’ （DFP）
- HessUpdate = ‘steepdesc’（最速下降方向）

注：新版已取消搜索步长选项（LineSearchType）

LargeScale: 'on' | 'off'

(on为缺省)

例5. $\min f(x,y)=100(y-x^2)^2+(1-x)^2$

精确解：$x=y=1$, $f(x,y)=0$

Exam0705.m

计算结果

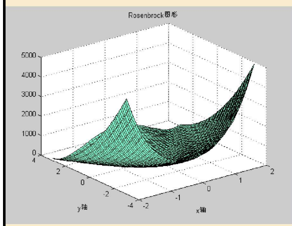
	搜索方向	步长搜索	最优解x_1	最优解x_2	最优值	目标函数调用次数
Case 1	bfgs	混合二、三次插值	9.9978e-001	9.9957e-001	4.7075e-008	1.5000e+002
Case 2	dfp		9.9950e-001	9.9899e-001	2.5440e-007	5.1300e+002
Case 3	Steepdesc		9.3080e-001	8.6673e-001	4.7999e-003	1.0020e+003
Case 4	bfgs	三次插值	9.9978e-001	9.9957e-001	4.7075e-008	1.5000e+002
Case 5	dfp		9.9950e-001	9.9899e-001	2.5440e-007	5.1300e+002
Case 6	Steepdesc		9.3080e-001	8.6673e-001	4.7999e-003	1.0020e+003

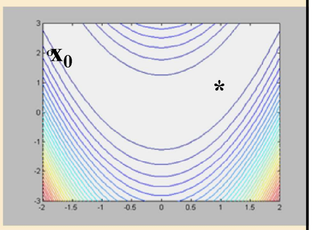
$f(x,y)=100(y-x^2)^2+(1-x)^2$ 的图形

Rosenbrock 函数

（香蕉函数）

exam0705plot.m





非线性最小二乘法：算法选择

$\min_{lb\le x\le ub} R(x)=r^T(x)r(x)$

Algorithm:'trust-region-reflective'(缺省)

| 'Levenberg-Marquardt'

当LargeScale = 'on'时尽可能用信赖域算法(需要用户提供梯度函数、Jacobi矩阵)

注：新版不提供Gauss-Newton法、步长选择

exam0705lsq.m

exam0705lsq_jacob_run.m

无约束优化：

模型：$\mathop{Min}_{x}\ f(x),\ x\in R^n$

采用分析梯度

```
GradObj='on'
x=fminunc(@fun,x0,opt)
```

fun.m中还要有$\nabla f(x)$

一般形式 `function [f,g]=fun(x)`

续例5. $\min f(x,y)=100(y-x^2)^2+(1-x)^2$

算出 $\nabla f=\begin{bmatrix}-400x(y-x^2)-2(1-x)\\ 200(y-x^2)\end{bmatrix}$

exam0705grad_run.m

计算结果

	搜索方向	步长搜索	最优解x_1	最优解x_2	最优值	目标函数调用次数
Case 1	bfgs	混合二、三次插值	9.9982e-001	9.9963e-001	3.4588e-008	5.0000e+001
Case 2	dfp		1.0002e+000	1.0003e+000	2.8158e-008	3.1800e+002
Case 3	Steepdesc		9.6446e-001	9.2976e-001	1.2808e-003	1.0000e+003
Case 4	bfgs	三次插值	9.9982e-001	9.9963e-001	3.4588e-008	5.0000e+001
Case 5	dfp		1.0002e+000	1.0003e+000	2.8158e-008	3.1800e+002
Case 6	Steepdesc		9.6446e-001	9.2976e-001	1.2808e-003	1.0000e+003

与不用分析梯度的结果比较

Case 1	bfgs	混合二、三次插值	9.9978e-001	9.9957e-001	4.7075e-008	1.5000e+002
Case 2	dfp		9.9950e-001	9.9899e-001	2.5440e-007	5.1300e+002
Case 3	Steepdesc		9.3080e-001	8.6673e-001	4.7999e-003	1.0020e+003
Case 4	bfgs	三次插值	9.9978e-001	9.9957e-001	4.7075e-008	1.5000e+002
Case 5	dfp		9.9950e-001	9.9899e-001	2.5440e-007	5.1300e+002
Case 6	Steepdesc		9.3080e-001	8.6673e-001	4.7999e-003	1.0020e+003

非线性最小二乘法：采用分析导数

$$r_i(x) = y_i - f(t_i, x)$$

$$r(x) = (r_1(x), \cdots r_n(x))^T$$

$$\min_{lb \le x \le ub} R(x) = r^T(x) r(x)$$

```
[x,resnorm,res,exitf,out,lambda,jacob]=
lsqnonlin(@fun,x0,lb,ub,options)
```

输入的用法与fminunc类似，但注意：fun.m ~r(x)的m文件名，Jacobian='on'时含有导数信息 $\nabla r(x)$

```
function [r,J] = fun(x)
```

几个值得注意的问题

梯度函数：利用分析梯度可能改进算法的性能

精度控制：对迭代次数有重大影响，应适当选择。

改变初始值 由一个初值出发通常得到局部最优解，如果函数存在多个局部最优，只有改变初值，对局部最优进行比较，才有可能得到全局最优解。

算法选择：BFGS公式，混合2，3次插值，一般较好。

其他算法选择： （详细用法请查阅help文档）

高度非线性、不连续时可用程序 fminsearch(@fun,x0)

单变量时可用程序 fminbnd(@fun,v1,v2)

优化工具箱的其他功能：非线性方程(组)

单变量方程求根（变号点）：$f(x) = 0, x \in R$

```
[x,fval,exitflag,output] =
fzero(fun,x0,options)
```

多变量方程组求零点： $F(x) = 0, x \in R^n$

```
[x,fval,exitflag,output,jacobian] =
fsolve(fun,x0,options)
```

优化工具箱的其他功能：非负最小二乘

$$\min_{x \ge 0} \|Cx - d\|_2^2$$

```
[x,resnorm,res,exitf,out,lambda]=
lsqnonneg(C,d,x0,options)
```

更一般地：带线性约束的线性最小二乘

$$\min_{lb \le x \le ub} \|Cx - d\|_2^2$$

$$s.t. \quad A_1 x \le b_1,\ A_2 x \le b_2, lb \le x \le ub$$

```
[x,resnorm,res,exitflag,output,lambda] =
lsqlin(C,d,A1,b1,A2,b2,lb,ub,x0,options)
```

无 约 束 优 化 实 验 内 容

实验目的

1）掌握 Matlab 优化工具箱的基本用法，对不同算法进行初步分析、比较。

2）练习用无约束优化方法建立和求解实际问题的模型（包括线性和非线性最小二乘拟合）。

内容 见网络学堂