

# 非线性数值优化

# 线性回归

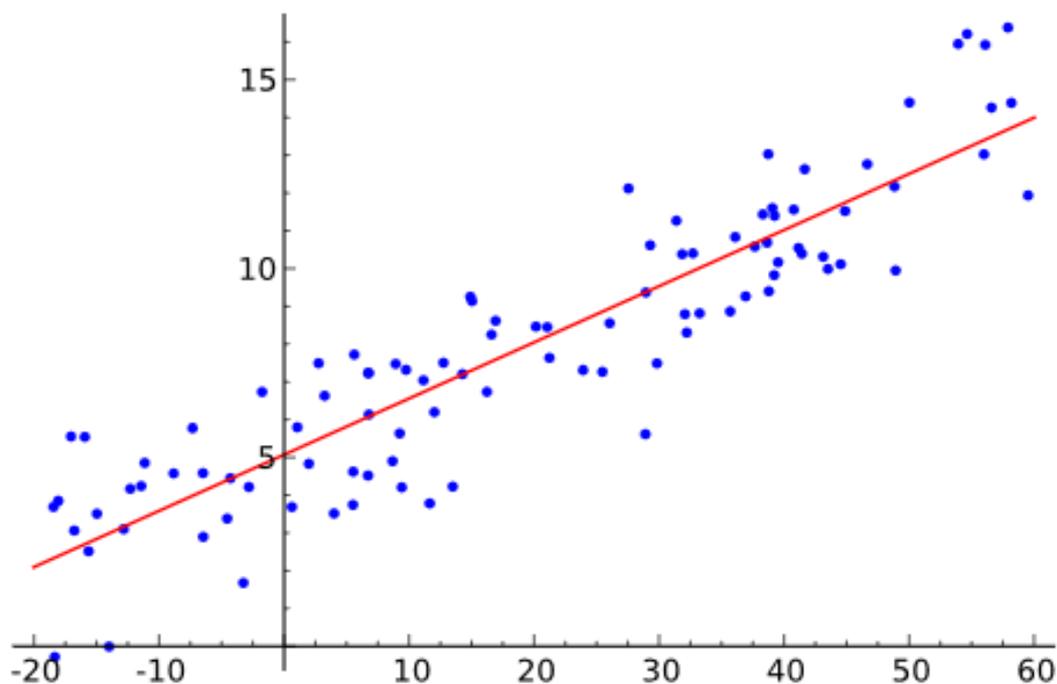
- 线性模型：

$$a^T x = b$$

- 输入，线性地确定，系数是

# 线性回归

- 给定了一系列观测值，如何确定？



# 最大似然估计

- 测量通常带有高斯噪音

$$b_i = a_i^T x + n, n \sim G(0, \sigma)$$

$$P[(a_i, b_i)|x] = P[b_i - a_i^T x] \propto \exp - \frac{(b_i - a_i^T x)^2}{2\sigma^2}$$



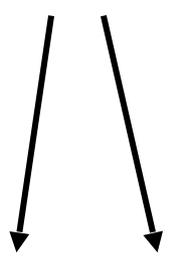
似然函数

# 最大似然估计

- 假定测量彼此独立

$$\begin{aligned} & P[(a_1, b_1)(a_2, b_2)\dots|x] \\ &= \prod_i P[(a_i, b_i)|x] \\ &= \prod_i P[b_i - a_i^T x] \\ &\propto \exp -\frac{\sum_i (b_i - a_i^T x)^2}{2\sigma^2} = \exp -\frac{\|Ax - b\|_2^2}{2\sigma^2} \end{aligned}$$

把纵向列写出来



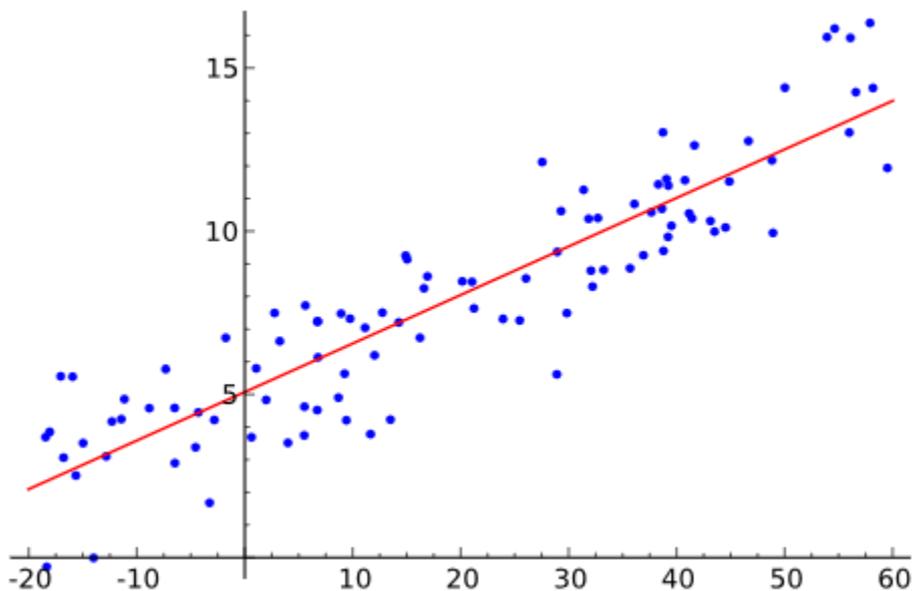
# 最大似然估计

- 最大似然估计寻找使似然概率最大化的

$$\begin{aligned}\hat{x} &= \arg \max_x P[(a_1, b_1)(a_2, b_2)\dots|x] \\ &= \arg \max_x \exp - \frac{\|Ax - b\|_2^2}{2\sigma^2} \\ &= \arg \min_x \|Ax - b\|_2^2\end{aligned}$$

# 线性最小二乘

- 由此可知线性最小二乘求解的是带高斯噪声的线性模型的拟合



# 非线性最小二乘

- 一般的非线性模型：

$$b = f_x(a)$$

- 我们定义函数

$$R(x) = \begin{pmatrix} b_1 - f_x(a_1) \\ \vdots \\ b_n - f_x(a_n) \end{pmatrix}$$

- 它代表了全体测量误差的向量

# 非线性最小二乘

- 在高斯噪音下对系数的最大似然估计对应于误差向量欧式范数的最小化：

$$\hat{x} = \arg \min_x \|R(x)\|_2^2$$

- 噪音依旧是高斯模型

# 病态问题

- 病态问题指的是条件缺少约束  
存在无穷多的解可以满足原问题
  - eg: 当方程数小于变量数
- 通过添加先验知识来约束解的性质

# 正则化

- 通过将正则化项加入最优化，可以避免出现病态解

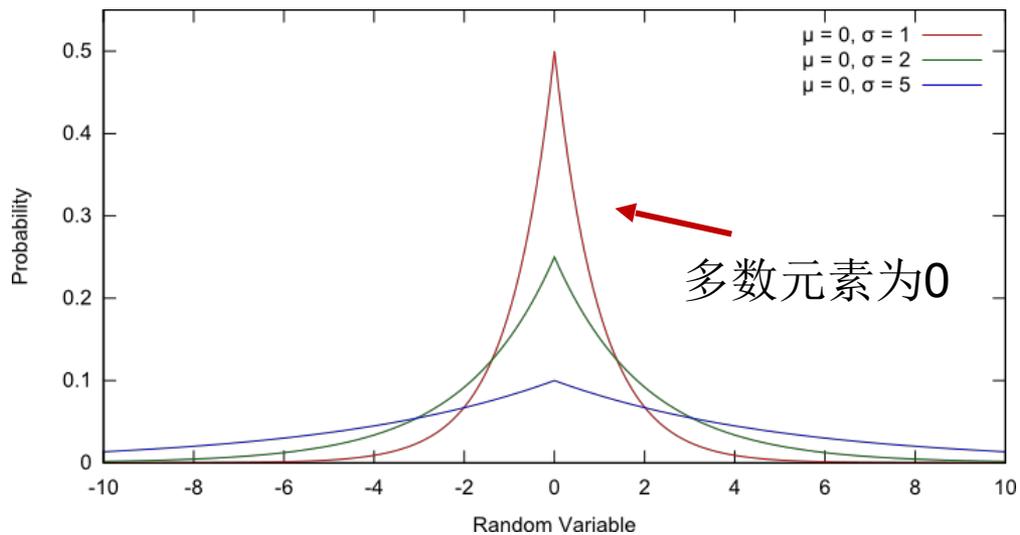
$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_1$$

# 正则化

- 一些正则化项对应了模型的先验概率
  - 假设的每个系数符合相同的Laplace分布

$$P(x_i) \propto \exp -\frac{|x_i|}{\sigma}$$



# 正则化

- 一些正则化项对应了模型的先验概率
  - 先验：的参数符合**Laplace**分布

$$P(x) = \prod_i P(x_i) \propto \exp -\frac{|x|_1}{r}$$

- 已知, 时的后验概率是

$$P(x|A, b) = \frac{P(A, b|x)P(x)}{P(A, b)}$$

# 正则化

- 在已知,条件下的最有可能的满足最大后验概率估计:

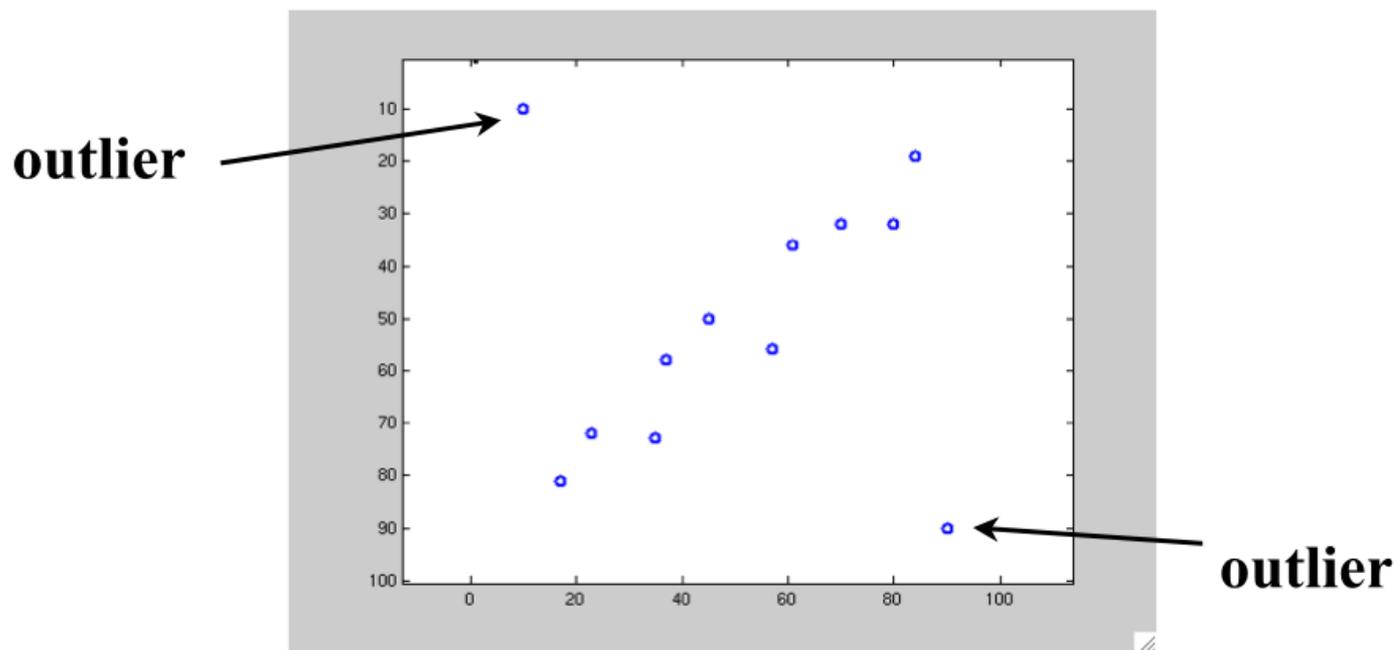
$$\begin{aligned}\arg \max_x P(x|A, b) &= \arg \max_x \frac{P(A, b|x)P(x)}{P(A, b)} \\ &= \arg \max_x P(A, b|x)P(x) \\ &= \arg \max_x \exp -\frac{\|Ax - b\|_2^2}{2\sigma^2} \exp -\frac{|x|_1}{r} \\ &= \arg \min_x \|Ax - b\|_2^2 + \frac{2\sigma^2}{r} |x|_1\end{aligned}$$

# 正则化

- 使用L1范数正则化
  - 倾向于非零系数更少的解
- 不同的正则化→不同的先验
- 带正则化的优化→最大后验概率估计

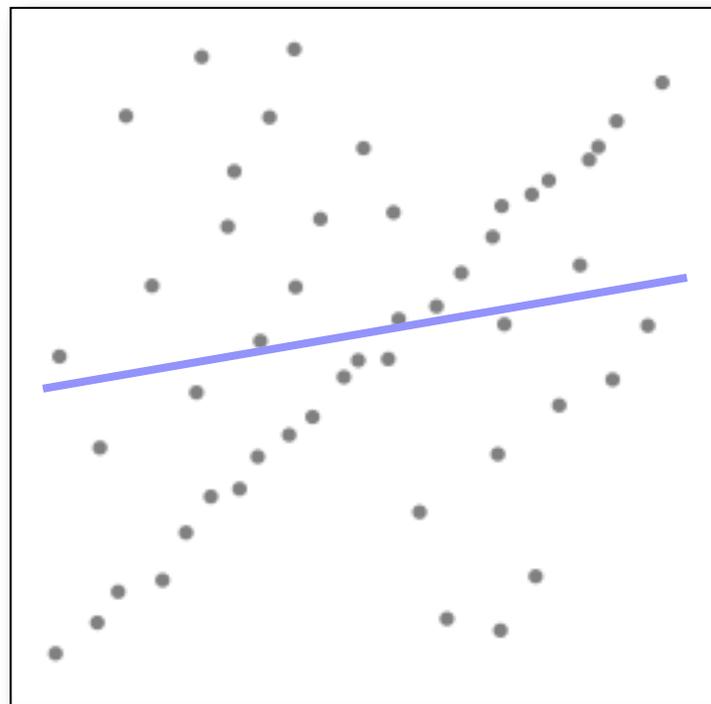
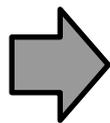
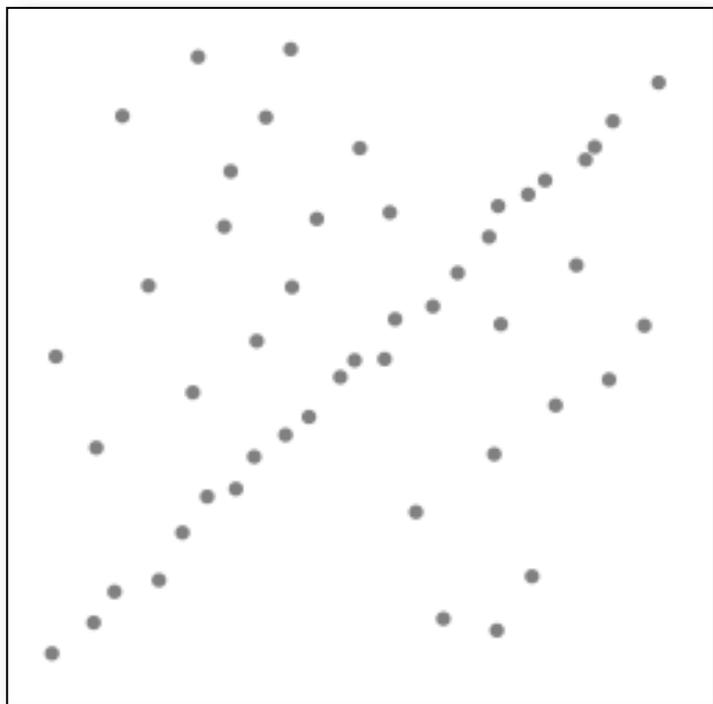
# Outlier

- Outlier = 不满足模型/噪音假设的错误点



# Outlier

- Outlier = 不满足模型/噪音假设的错误点



# Outlier

- 错误点的存在会严重影响模型拟合的结果
  - 是否可以减少错误点带来的影响？

# 鲁棒最小二乘

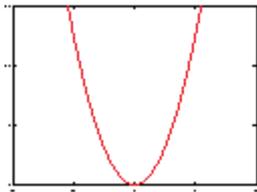
- 如果点很好地满足模型，认为是inlier， 否则是outlier
  - 对于outlier，我们减少其误差带来的权重
  - M估计

# M估计

- 引入鲁棒函数代替

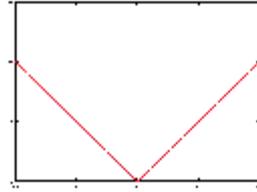
# M估计

Least-squares



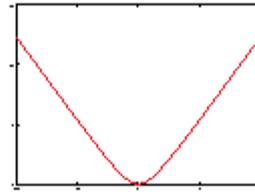
$\rho$ -function

Least-absolute



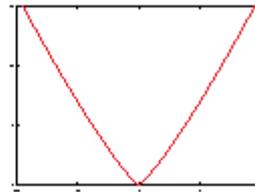
$\rho$ -function

$L_1 - L_2$



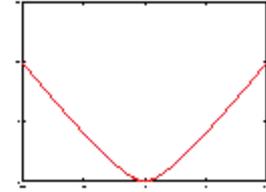
$\rho$ -function

Least-power



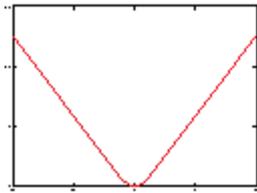
$\rho$ -function

Fair



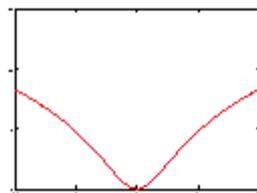
$\rho$ -function

Huber



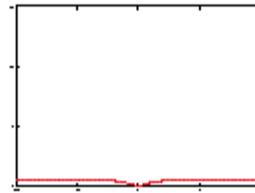
$\rho$ -function

Cauchy



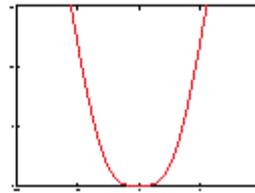
$\rho$ -function

Geman-McClure



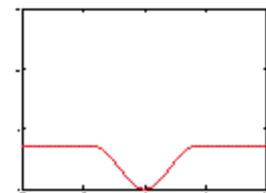
$\rho$ -function

Welsch



$\rho$ -function

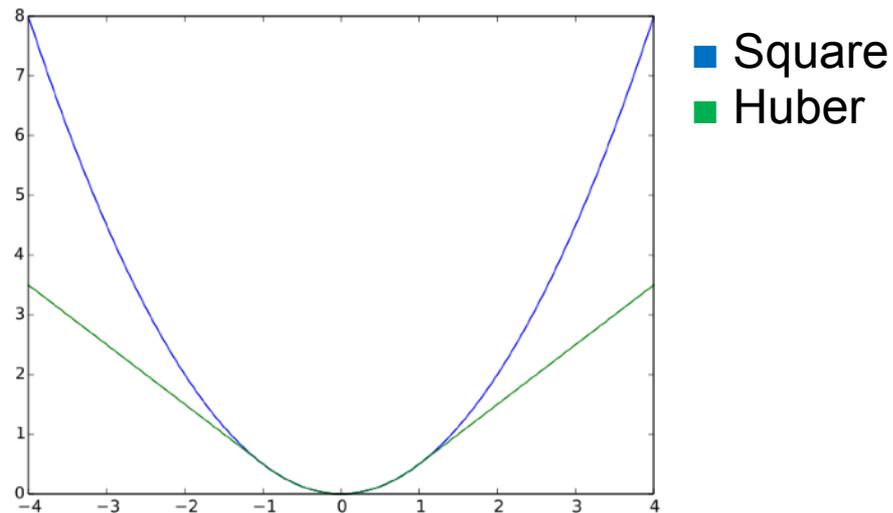
Tukey



$\rho$ -function

# M估计

- Huber function: 大误差使用线性距离



$$\rho(x) = \begin{cases} x^2/2 & |x| \leq k \\ k(|x| - k/2) & |x| > k \end{cases}$$

# 鲁棒估计

- 除了M估计，还有各种鲁棒过程
  - RANSAC: Random Sample Consensus

## Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:

- s** — the smallest number of points required
- N** — the number of iterations required
- d** — the threshold used to identify a point that fits well
- T** — the number of nearby points required  
to assert a model fits well

Until **N** iterations have occurred

Draw a sample of **S** points from the data  
uniformly and at random

Fit to that set of **S** points

For each data point outside the sample

Test the distance from the point to the line  
against **d** if the distance from the point to the line  
is less than **d** the point is close

end

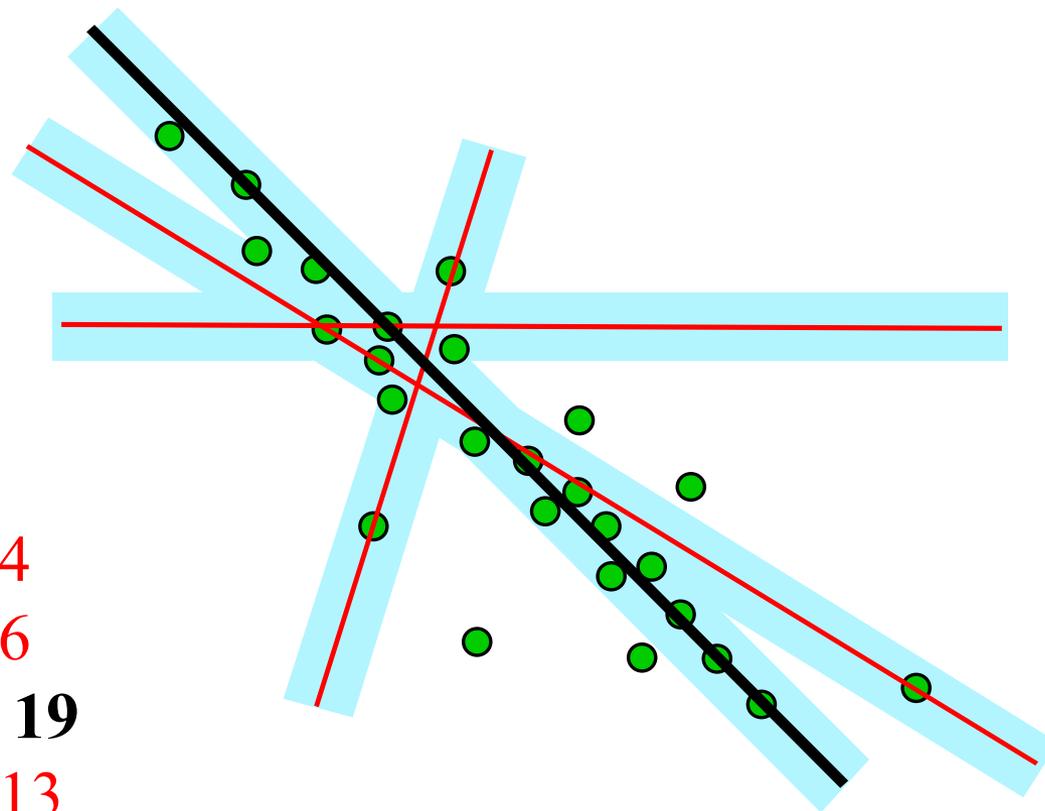
If there are **T** or more points close to the line  
then there is a good fit. Refit the line using all  
these points.

end

Use the best fit from this collection, using the  
fitting error as a criterion

(Forsyth & Ponce)

# Ransac Procedure

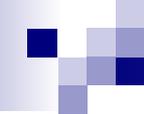


Count = 4

Count = 6

**Count = 19**

Count = 13



# 数值优化方法

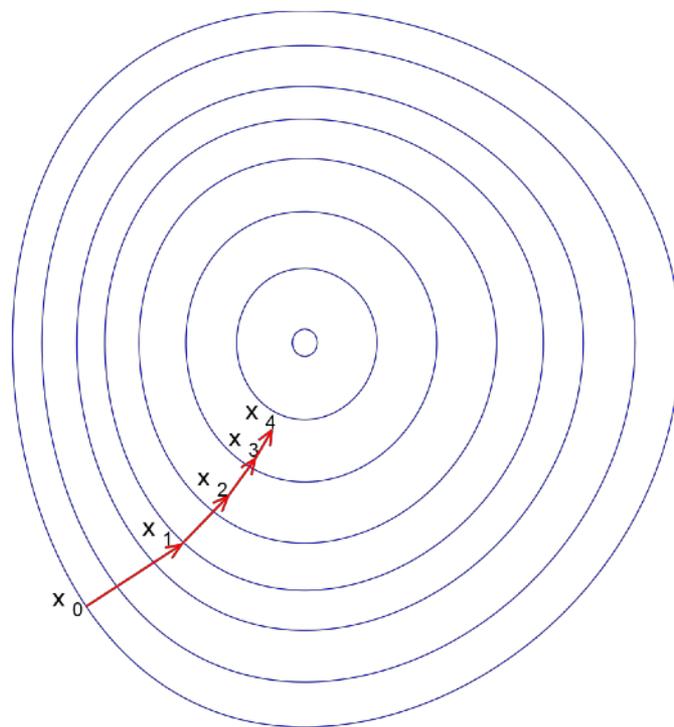
# 下降方法

- 常见的数值优化算法通过寻找一系列使目标函数不断下降的变量值完成优化

$$F(x_0) > F(x_1) > \dots > F(x_k) > \dots$$



# 下降方法



# 下降方法

- while not converge

# 下降方向的确定

- 观察目标函数在的一阶**Taylor**展开

$$F(x_0 + \Delta x) \approx F(x_0) + J_F \Delta x$$

- 当时\*, 函数的值会下降

\*要足够小

# 最速下降法

- 何时下降速度最快
  - 当的方向与相同时

# 最速下降法

- while not converge

# 最速下降法

- 步长如何确定？
  - 一维直线搜索
    - 精确搜索
    - 近似搜索

## Line search types

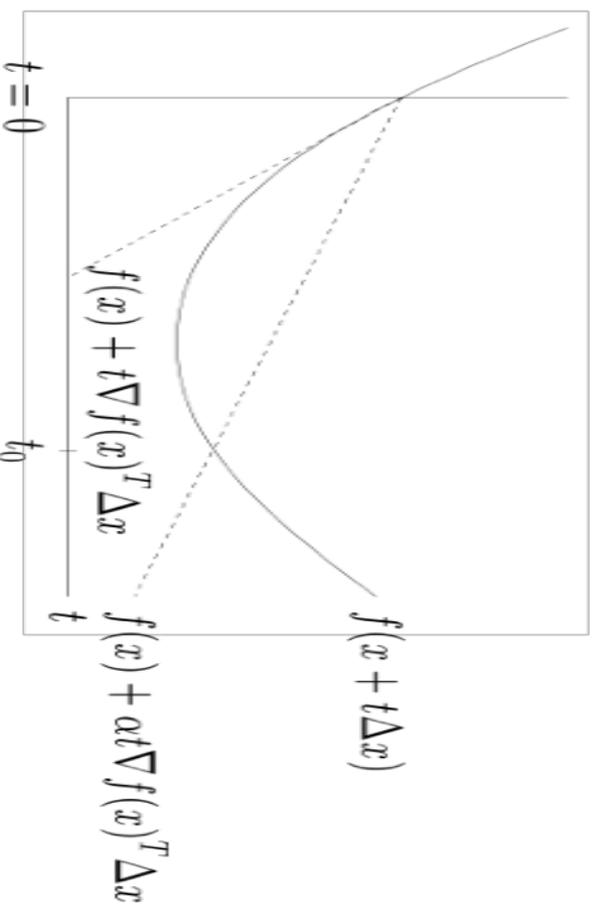
**exact line search:**  $t = \operatorname{argmin}_{t > 0} f(x + t\Delta x)$

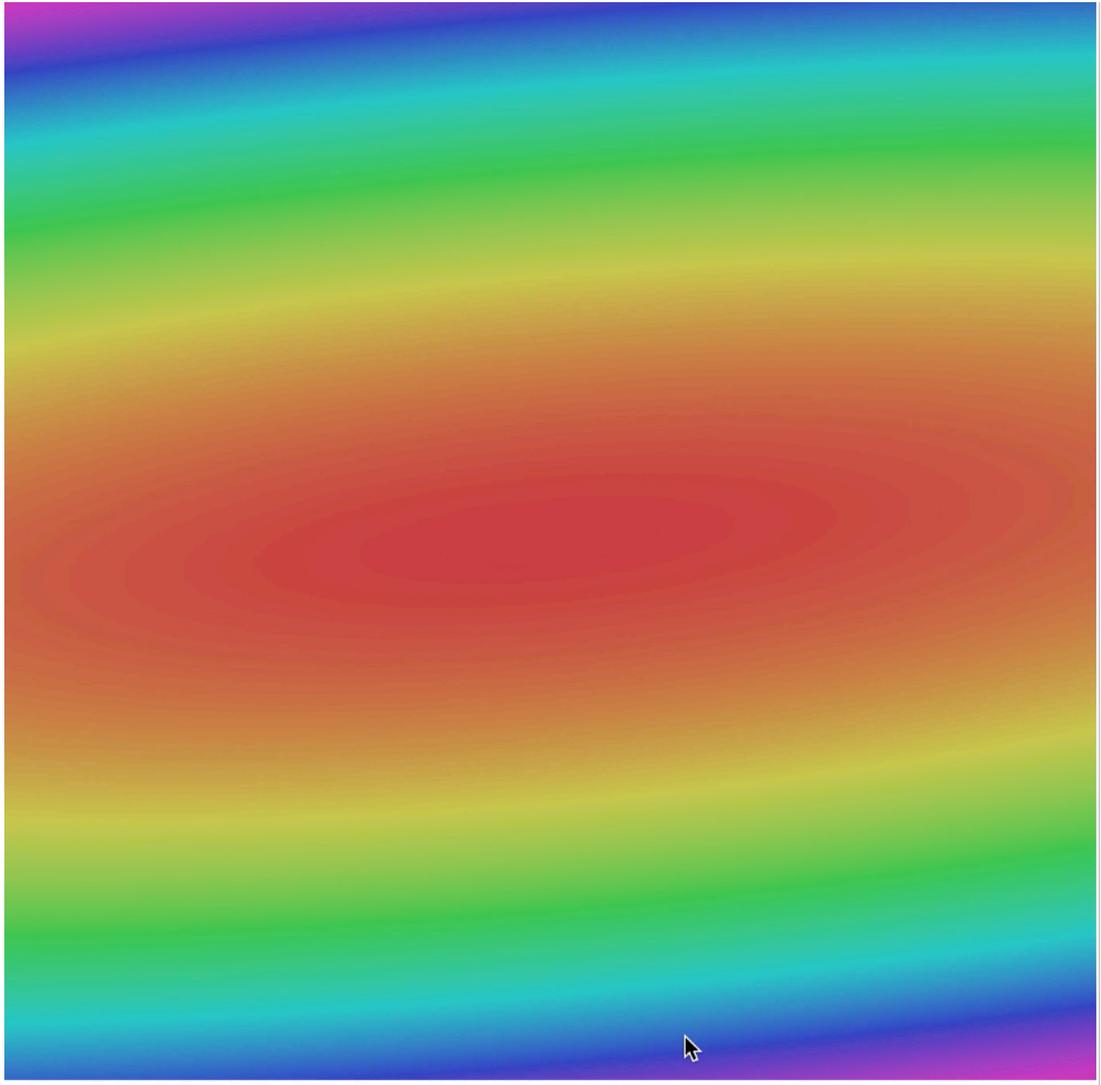
**backtracking line search** (with parameters  $\alpha \in (0, 1/2)$ ,  $\beta \in (0, 1)$ )

- starting at  $t = 1$ , repeat  $t := \beta t$  until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until  $t \leq t_0$

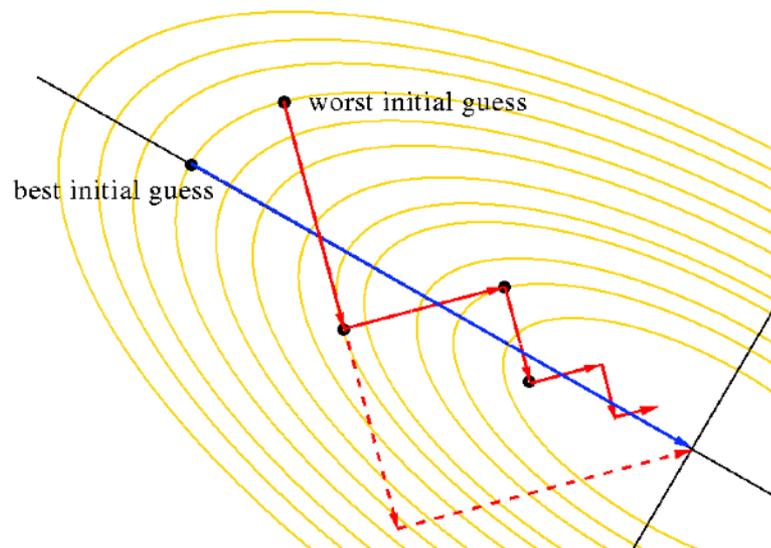
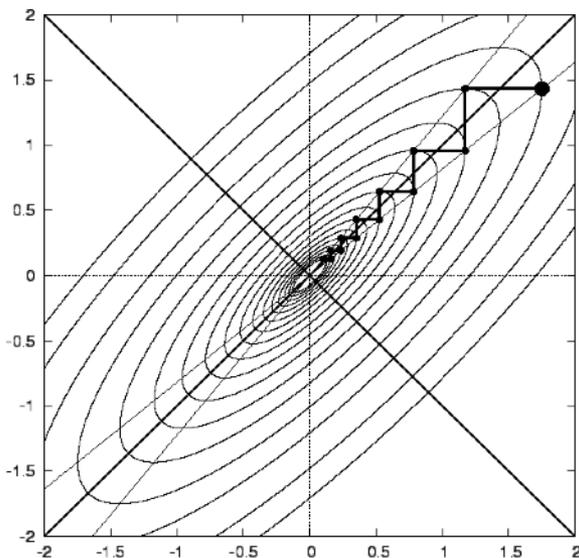




# 最速下降法

- 特点:

- 精确线性搜索时，下降方向彼此正交
- 但如果初值不好.....



# 最速下降法

- 优点：
  - 实现简单计算量少
  - 在距离最优值较远时往往表现很好（启动快）
- 缺点：
  - 在最优值附近收敛缓慢
  - 当能量函数性质不好时会浪费很多迭代

# 共轭梯度法

- 给定一个二次型

$$f(x) = \frac{1}{2}x^T Ax + bx$$

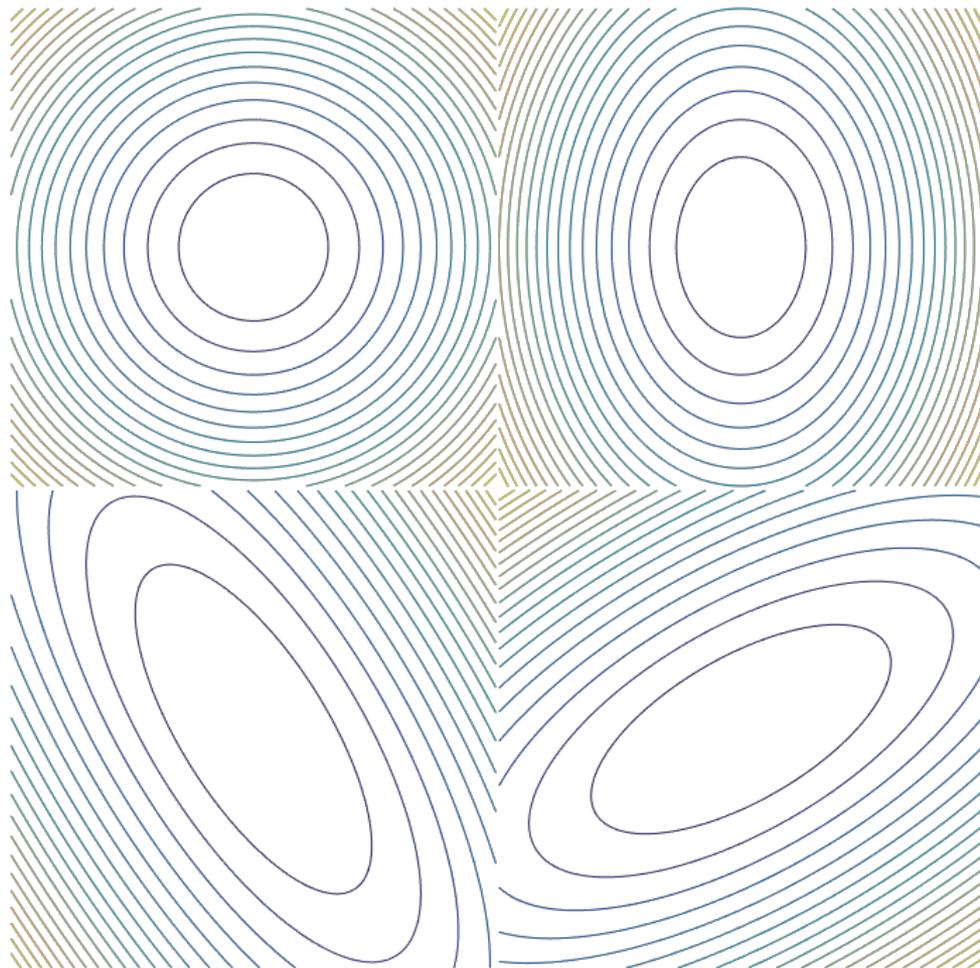
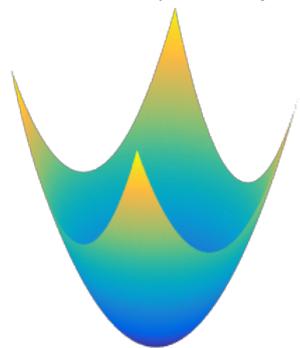
对称



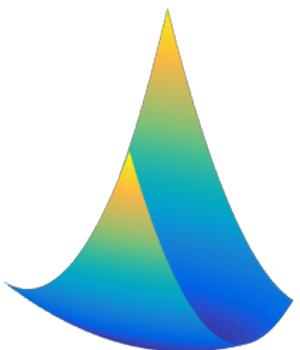
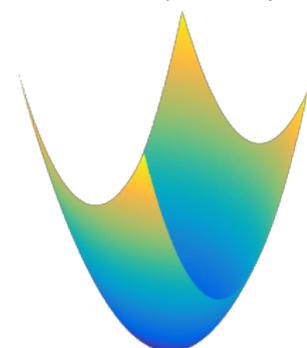
- 它的形状是什么样的？

# 共轭梯度法

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$



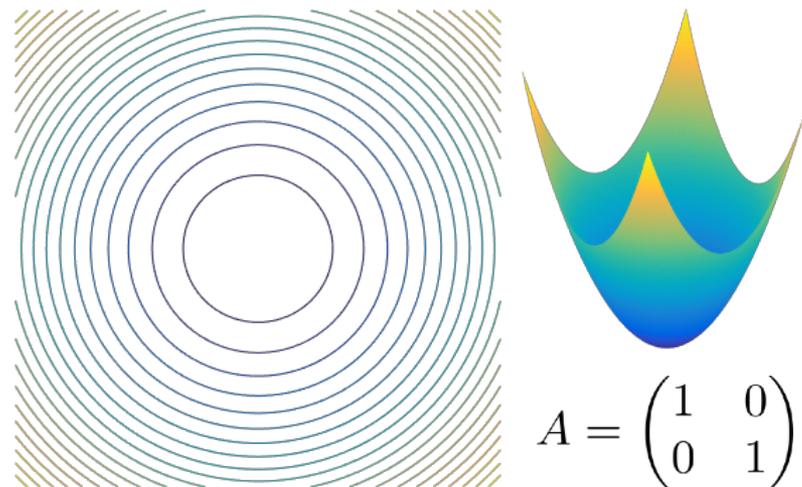
$$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 2 \end{pmatrix}$$



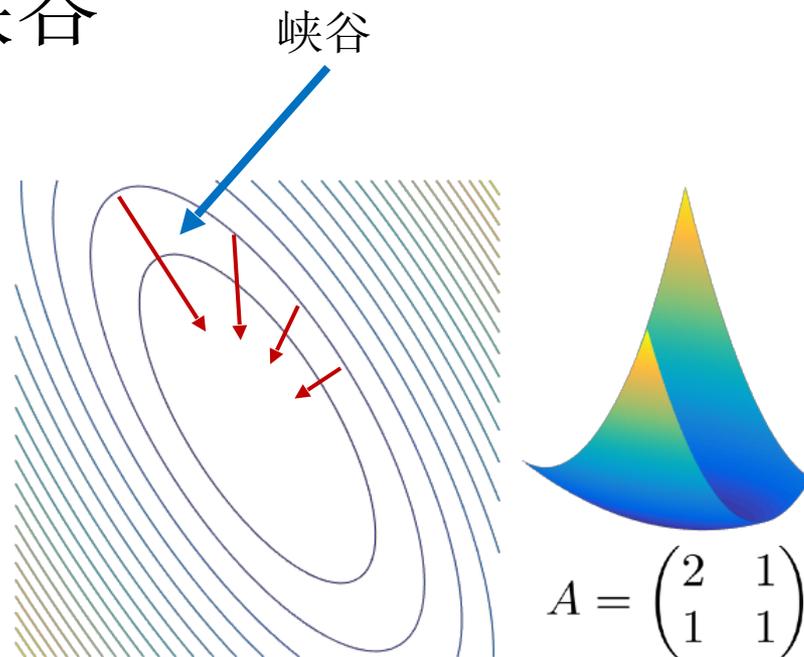
# 共轭梯度法

- 如果局部很均匀，也就不存在峡谷和峭壁



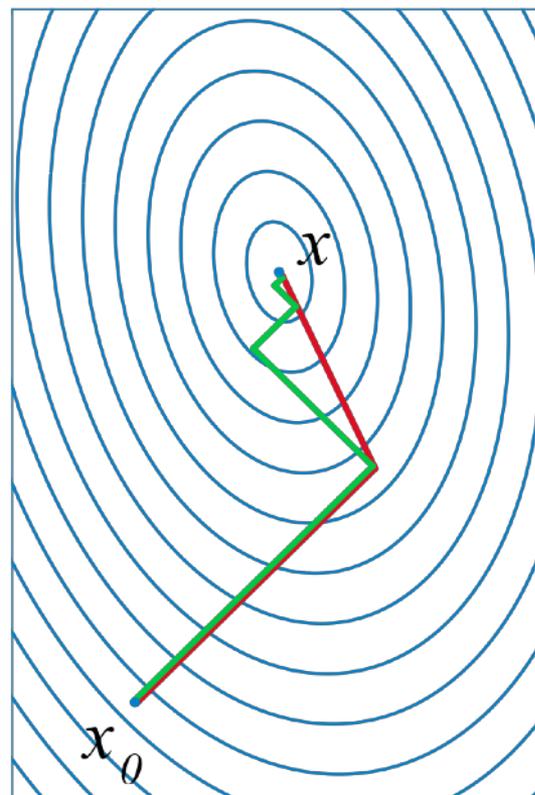
# 共轭梯度法

- 包含了局部的曲率信息  
我们可以知道哪里有峡谷



# 共轭梯度法

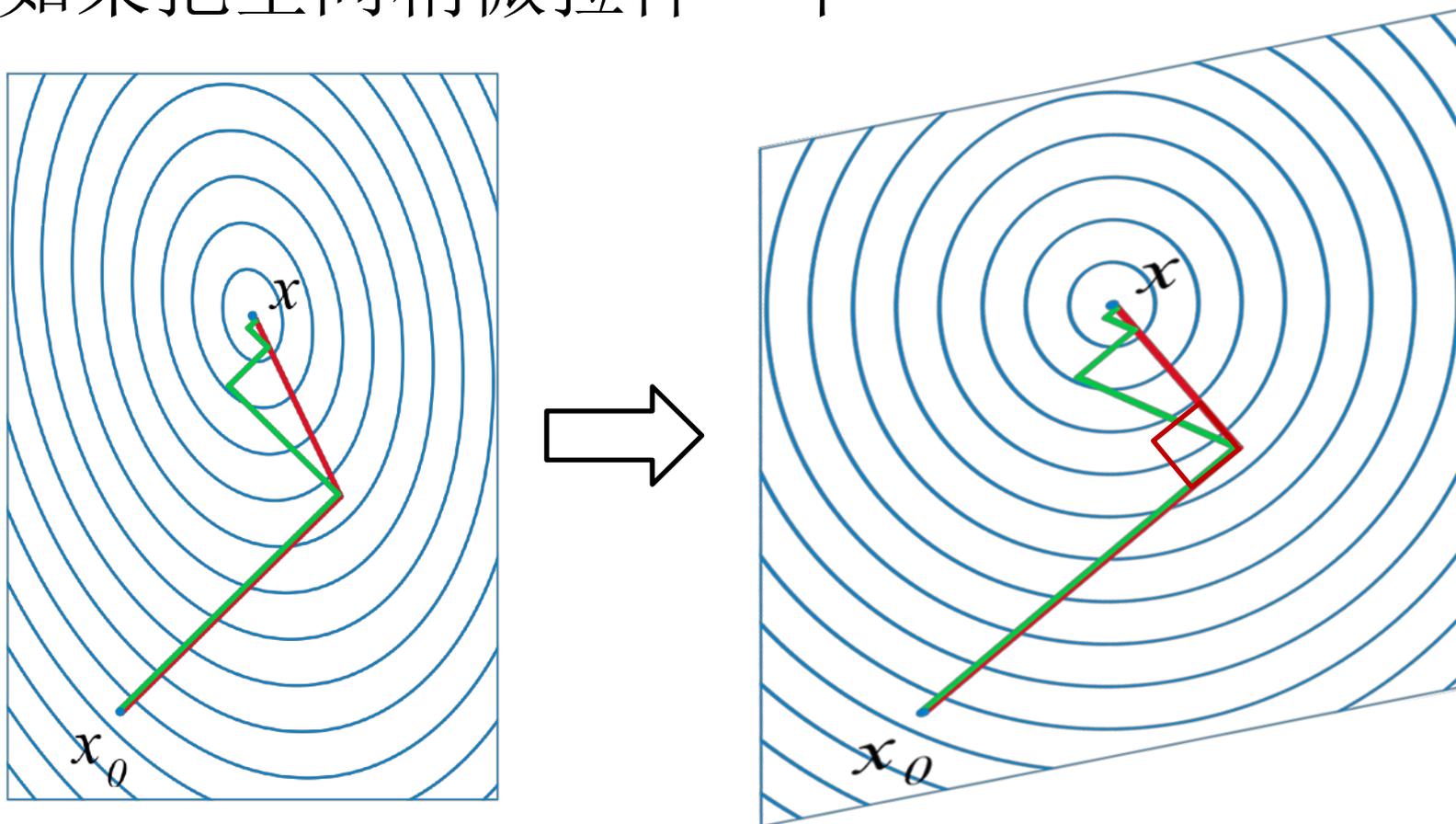
- 利用局部曲率信息  
我们可以沿着狭长的峡谷更快下降



■ 最速下降    ■ 共轭梯度

# 共轭梯度法

- 如果把空间稍微拉伸一下



# 共轭梯度法

- 最速下降法中
  - 下降方向互相垂直 = 各个方向同等对待
- 共轭梯度法
  - 下降方向互相“共轭”
  - 共轭 = 如果把峡谷拉伸均匀，此时垂直

# 共轭梯度法

- 寻找共轭
  - 利用矩阵诱导内积

正交	共轭
$\langle x, y \rangle = x^T y = 0$	$\langle x, y \rangle_A = x^T A y = 0$

# 共轭梯度法

- 二次型的问题第3次课已经讲过
  - 反复求新的共轭方向并迭代
- 一般的函数呢？
  - 用二次型近似
  - 二阶Taylor展开

$$F(x_k + \Delta x) \approx F(x_k) + J_F \Delta x + \frac{1}{2} \Delta x^T H_F \Delta x$$

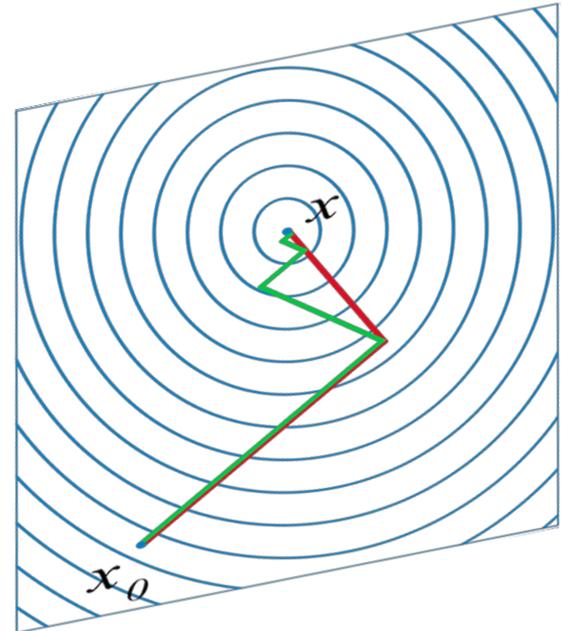


我们要的

# 共轭梯度法（非线性）

- // 初始方向采用负梯度方向
- while not converge

用函数的Hessian诱导共轭



# 共轭梯度法

- 事实上共轭方向的计算并不需要**Hessian**
  - 自己尝试推导

<http://andrew.gibiansky.com/blog/machine-learning/hessian-free-optimization/>

# 共轭梯度法

- 优点：
  - 实现简单
  - 启动快
  - 比最速下降法稳定
- 缺点：
  - 最优值附近收敛慢
  - 什么方法能快速收敛？
    - Newton-Rapson法 →

# Newton-Rapson

- 继续回到二次型

$$f(x) = \frac{1}{2}x^T Ax + bx$$

对称正定



- 它的最小值是解
  - 可以用线性共轭梯度求解

# Newton-Rapson

- 任意目标函数呢？
  - 用二次型近似
  - 二阶Taylor展开

# Newton-Rapson

- 在附近进行二阶展开

$$F(x_k + \Delta x) \approx F(x_k) + J_F \Delta x + \frac{1}{2} \Delta x^T H_F \Delta x$$

- 求解最小化的

$$H_F \Delta x + J_F^T = 0$$

- 迭代更新

$$x_{k+1} = x_k + \gamma \Delta x = x_k - \gamma H_F^{-1} J_F^T$$

# Newton-Rapson

- while not converge

↑  
牛顿步

# Newton-Rapson

- 优点：收敛快
  - 在极值的邻域内二次收敛
  - 适合接近最终结果时使用
- 缺点：Hessian 计算量很大
  - 有时无法计算
  - 能不能近似？
    - Gauss-Newton →

# Gauss-Newton

- 利用问题的性质
  - 最小二乘问题

$$F(x_k + \Delta x) = \|R(x_k + \Delta x)\|_2^2$$

# Gauss-Newton

- 按照NR方法展开左侧

$$F(x_k + \Delta x) \approx F(x_k) + J_F \Delta x + \frac{1}{2} \Delta x^T H_F \Delta x$$

- 将右侧进行一阶近似展开

$$\begin{aligned} \|R(x_k + \Delta x)\|_2^2 &\approx \|R(x_k) + J_R \Delta x\|_2^2 \\ &= \underbrace{\|R(x_k)\|_2^2}_{F(x_k)} + \underbrace{2R(x_k)^T J_R}_{J_F} \Delta x + \Delta x^T J_R^T J_R \Delta x \end{aligned}$$

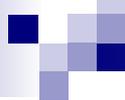
# Gauss-Newton

- 两侧简单比较能得到

$$H_F \approx 2J_R^T J_R$$

- 此时最优的满足

$$J_R^T J_R \Delta x + J_R^T R(x_k) = 0$$



# Gauss-Newton

- while not converge

# Gauss-Newton

- 优点
  - 不需要Hessian，容易计算
  - 收敛快
- 缺点
  - 如果不可逆呢？

# Levenberg-Marquardt

- 如果不可逆，GN会变得不稳定

$$J_R^T J_R \Delta x + J_R^T R(x_k) = 0$$

↑  
有无穷多解

# Levenberg-Marquardt

- LM法通过“正则化”回避这个问题

$$(J_R^T J_R + \lambda I) \Delta x + J_R^T R(x_k) = 0$$

# Levenberg-Marquardt

$$(J_R^T J_R + \lambda I) \Delta x + J_R^T R(x_k) = 0$$

- 对于全部  $\lambda$  ， 一定是正定的

# Levenberg-Marquardt

$$(J_R^T J_R + \lambda I) \Delta x + J_R^T R(x_k) = 0$$

- 的效果
  - : 梯度下降步, 并且长度短
  - : Gauss-Newton 步

# Levenberg-Marquardt

- 的选择
  - 每轮迭代更新
  - 当下降明显时,
  - 当下降不明显时,

# Levenberg-Marquardt

- while not converge

← 不再使用任何线性搜索

# Levenberg-Marquardt

- L-M算法不再需要任何线性搜索
  - 反应了当前点逼近二次型的程度
  - 它估计二次型的逼近效果，直接寻找更优点
- 这类算法称为信赖域算法

# Levenberg-Marquardt

- 优点：
  - 启动快 ( )
  - 收敛快 ( )
  - 不退化 (总是正定)
  
- $LM = SD + GN$

# 思考

- 什么情况下不可逆？
- 带正则化最小二乘和鲁棒最小二乘如何使用GN/LM算法？

# 参考资料

- Zhengyou Zhang, Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting.
  - <http://research.microsoft.com/en-us/um/people/zhang/inria/publis/tutorial-estim/Main.html>
- J.R. Shewchuk, An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.
  - <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>
- K. Madsen, H.B. Nielsen, O. Tingleff, Methods for Non-Linear Least Squares Problems.
  - [http://www2.imm.dtu.dk/pubdb/views/edoc\\_download.php/3215/pdf/imm3215.pdf](http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3215/pdf/imm3215.pdf)