非线性数值优化

计算摄影学第五讲

数学优化（最优化）

- (Mathematical) Optimization

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & g_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

- $x \in \mathbf{R}^n$ is (vector) variable to be chosen
- f_0 is the *objective function*, to be minimized
- $f_1, \ldots, f_m$ are the *inequality constraint functions*
- $g_1, \ldots, g_p$ are the *equality constraint functions*

最大化问题可以转化为最小化问题

https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

数学优化

- 为什么要学习最优化?
  - 图像处理问题大多可以描述为最优化问题
  - 最优化是机器学习/深度学习的基础
  - 方法论：几乎任何决策问题都可以建模成数学优化问题来求解

- 如何把问题建模成最优化问题?
  - 确定优化变量
  - 制定目标函数
  - 写出限制条件

Example

某军工厂生产甲、乙、丙三种产品，生产三 种产品需要A、B两种资源，其单位需求量及利润由下表1给出，问每天生产甲、乙、丙三种产品各多少，可使总利润最大?

	甲	乙	丙	资源的最大量
A	2	3	1	100kg
B	3	3	2	120kg
利润	40元	45元	24元	

$$max_{x_1,x_2,x_3} 40x_1 + 45x_2 + 24x_3$$

$$s.t.\begin{cases} 2x_1 + 3x_2 + x_3 \le 100 \\ 3x_1 + 3x_2 + 2x_3 \le 120 \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0 \end{cases}$$

Example

- Poisson image editing

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

线性回归问题

- 线性模型：

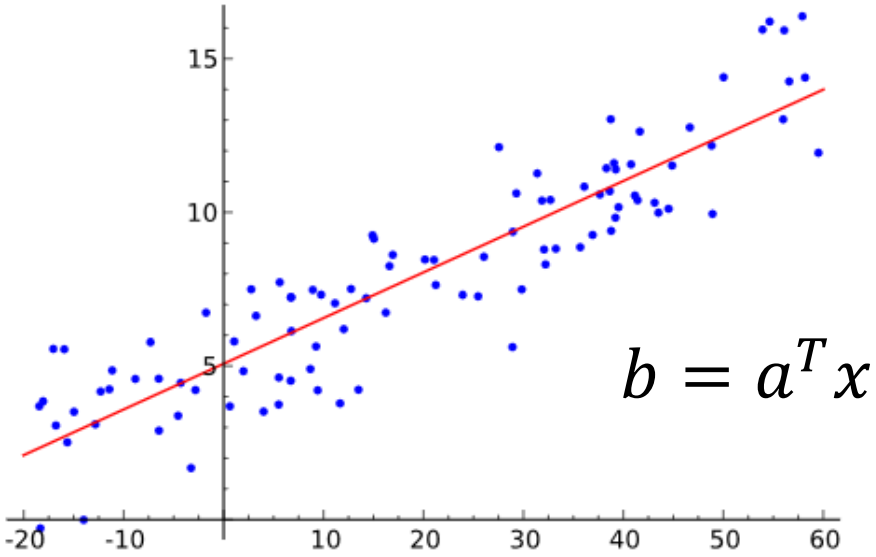
$$b = a^T x$$

输入a，输出b，系数是x

- 例如：
  - a=学习时间, b=考试成绩
  - a=确诊人数, b=疫情死亡人数

线性回归问题

- 给定了一系列观测值(a_i, b_i)，如何确定x?



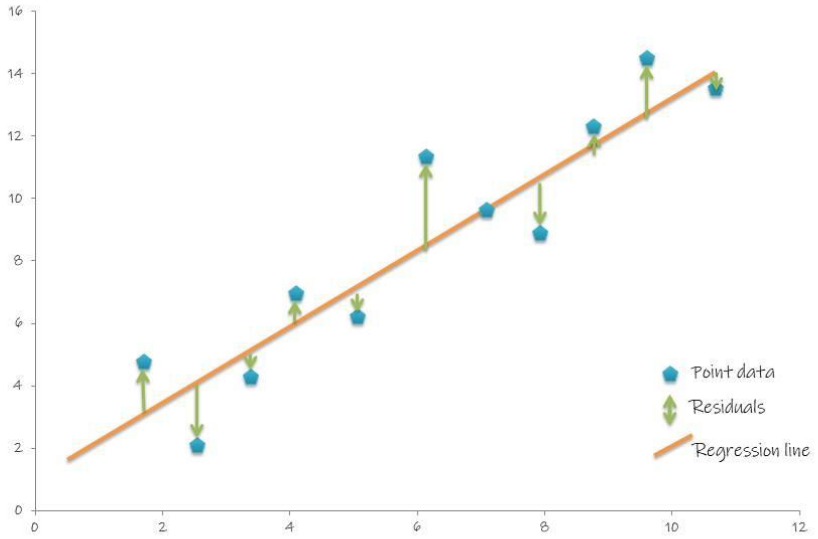
- 从数据中估计模型参数一般称为模型拟合（model fitting）或者回归（regression）

最小二乘法

- **最小二乘法**：最小化**均方误差**（Mean Square Error， 简称MSE）

$$\hat{x} = \arg\min_{x} \sum_{i} (b_i - a_i^T x)^2$$

- **残差**： $r_i = b_i - a_i^T x$
- **残差向量**：$R = [r_1, \cdots, r_n]^T$



L2范数

L2范数：$\| x \|_2 = \sqrt{\sum_i {x_i}^2}$

- 向量x的欧式距离

MSE 等于残差向量 $R = [r_1, \cdots, r_n]^T$ 的L2范数的平方

$$MSE = \| R \|_2^2$$

最小二乘的统计解释

- 假设数据带有高斯噪音

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

- 给定x观察到(a_i, b_i)的可能性：

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

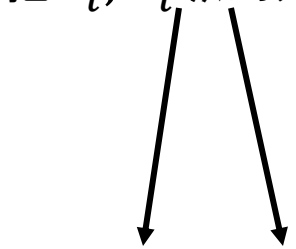
- P通常被称为：**似然函数 / 似然性（likelihood）**

最小二乘的统计解释

- 假定观测数据点彼此独立，则联合似然函数如下

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

把a_i, b_i纵向列写出来



最大似然估计

- 最大似然估计寻找使似然函数最大化的x

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

- 最小二乘对应于高斯噪声假设下的最大似然估计

非线性最小二乘

- 一般的非线性模型：$b = f_x(a)$

- 定义残差

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

- 最大似然估计等效于最小化MSE

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

病态问题

- ex： $Ax = b$当方程数小于变量数
  - 欠定问题

$$\min_{x} |x|_1$$
$$\text{s.t. } Ax = b$$

数值优化方法

如何求解最优化问题?

- 有些问题有解析解

  例如：线性最小二乘

$$\hat{x} = \arg\min_{x} \| Ax - b \|_2^2$$

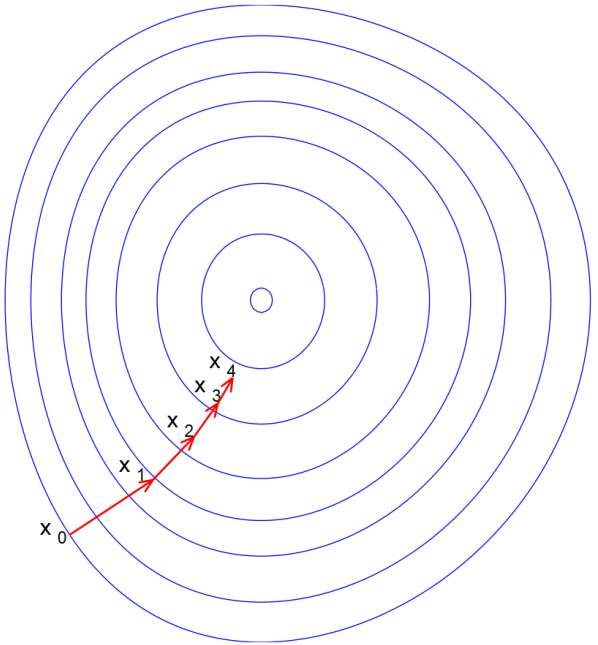
  的解为以下线性方程组的解（目标函数关于x求导为0）

$$A^T A x = A^T b$$

如何求解最优化问题?

- 大部分问题没有解析解的问题
- 通用策略：寻找一系列使目标函数不断下降的变量值完成优化

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



梯度下降

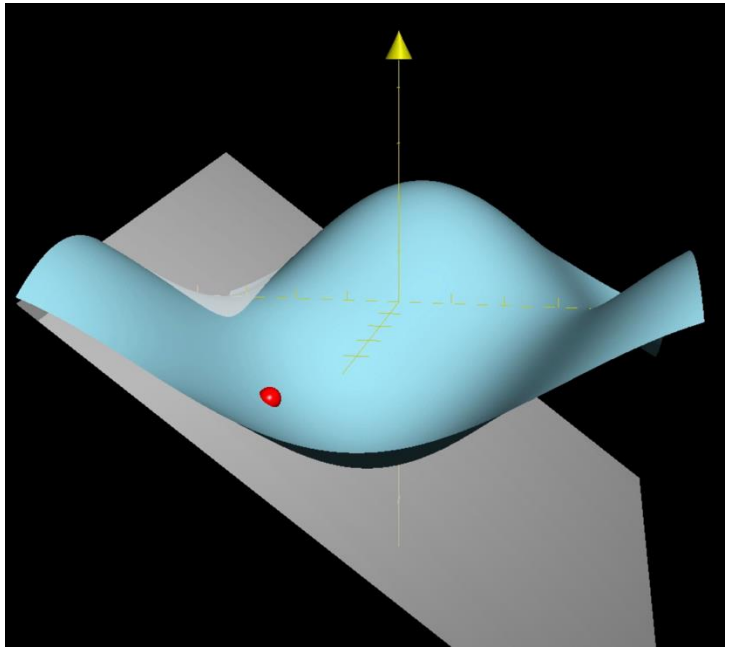
- $x \leftarrow x_0$ %初始化
- while not converge
  - $p \leftarrow \text{descending_direction}(x)$ %确定下降方向
  - $\alpha \leftarrow \text{descending_step}(x, p)$ %确定步长
  - $x \leftarrow x + \alpha p$ %更新变量

- *如何确定下降方向与步长?*

Preliminary: 多变量函数求导与逼近

- 一阶近似

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



$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix}$$

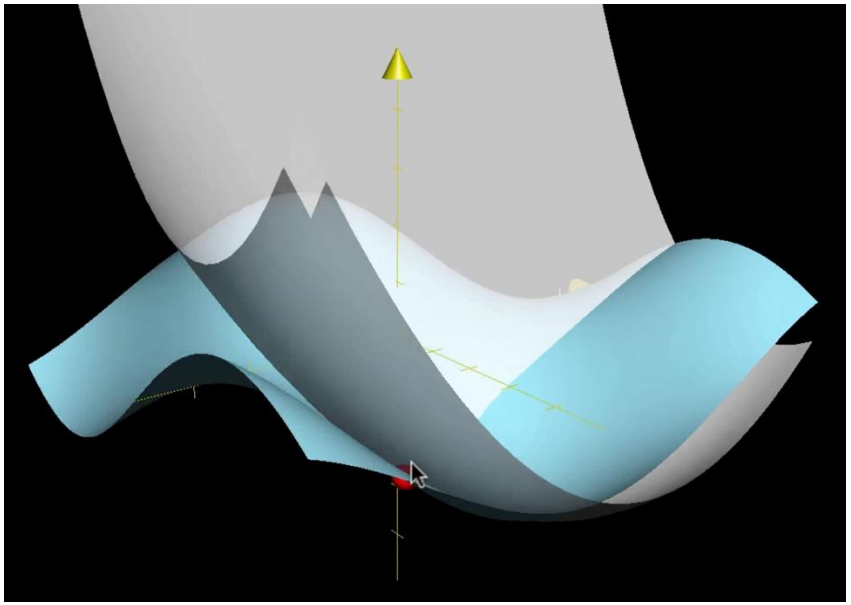
雅克比矩阵

https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/quadratic-approximations/v/what-do-quadratic-approximations-look-like

Preliminary: 多变量函数求导与逼近

- 二阶近似

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



$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \, \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \, \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \, \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \, \partial x_1} & \frac{\partial^2 f}{\partial x_n \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

海森矩阵

https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/quadratic-approximations/v/what-do-quadratic-approximations-look-like

下降方向的确定

- 对目标函数$F(x)$在x_0上进行一阶Taylor展开

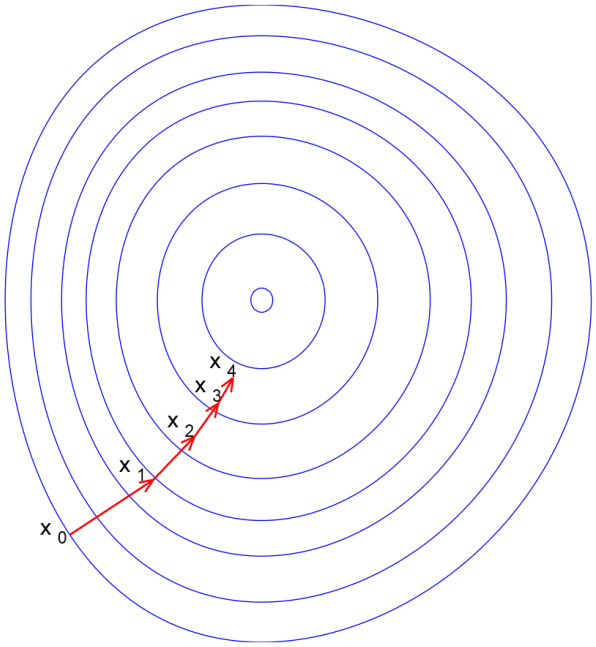
$$F(x_0 + \Delta x) \approx F(x_0) + J_F \Delta x$$

- 当$J_F \Delta x < 0$时，
  函数的值会下降（Δx要足够小）

最速下降法

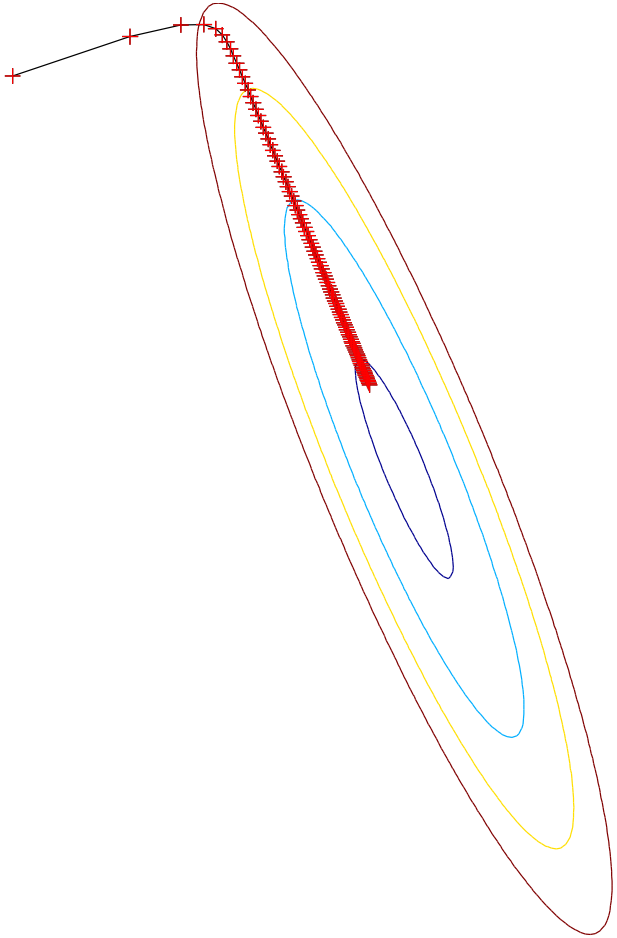
- $F(x_0 + \Delta x)$何时下降速度最快
  - 当Δx的方向与$-J_F^T$（负梯度方向）相同时

- 算法：
  - $x \leftarrow x_0$
  - while not converge
    - $x \leftarrow x - \alpha J_F^T$

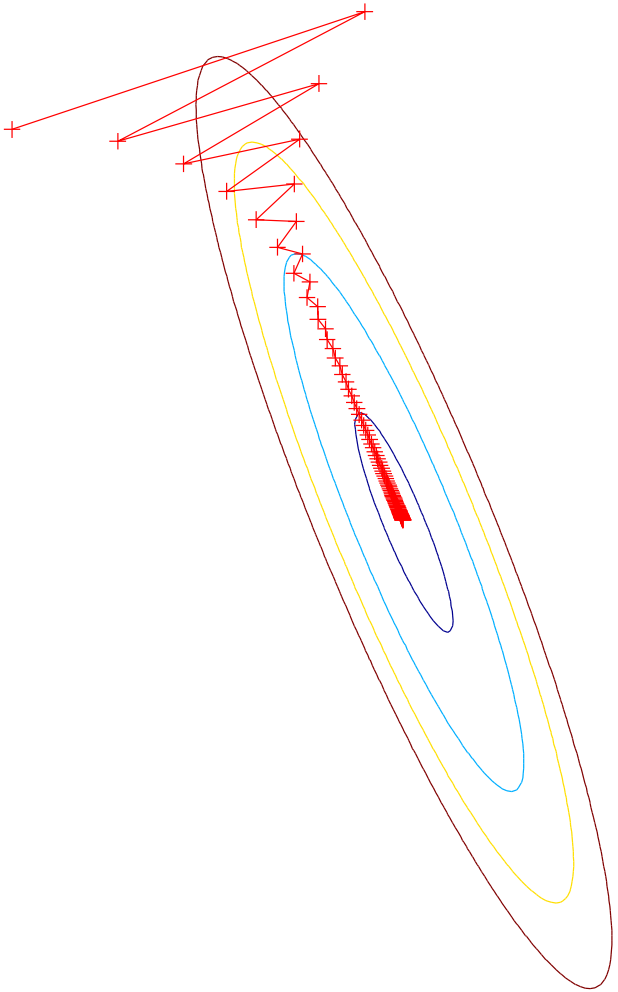


梯度方向垂直于等高线

步长的影响



α 太小



α 太大

步长的确定

确定h之后F是α的函数　　$\varphi(\alpha) = F(\mathbf{x}+\alpha\mathbf{h})\,, \quad \mathbf{x}$ and $\mathbf{h}$ fixed, $\alpha \geq 0\,.$

α 太小，函数值变化太小

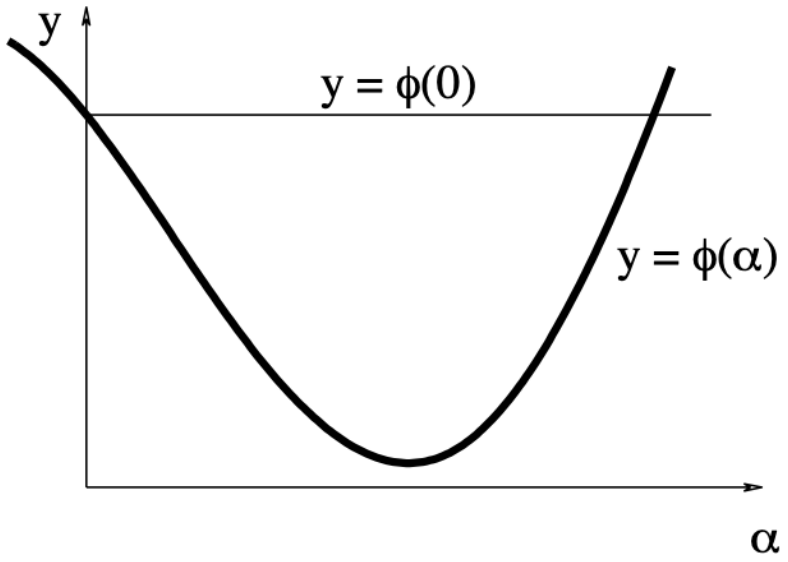
— 需要增大α

α 太大，$\phi(\alpha) > \phi(0)$

— 需要减小α

α 接近于$\phi(\alpha)$的minimizer

— 可接受的α值



步长的确定

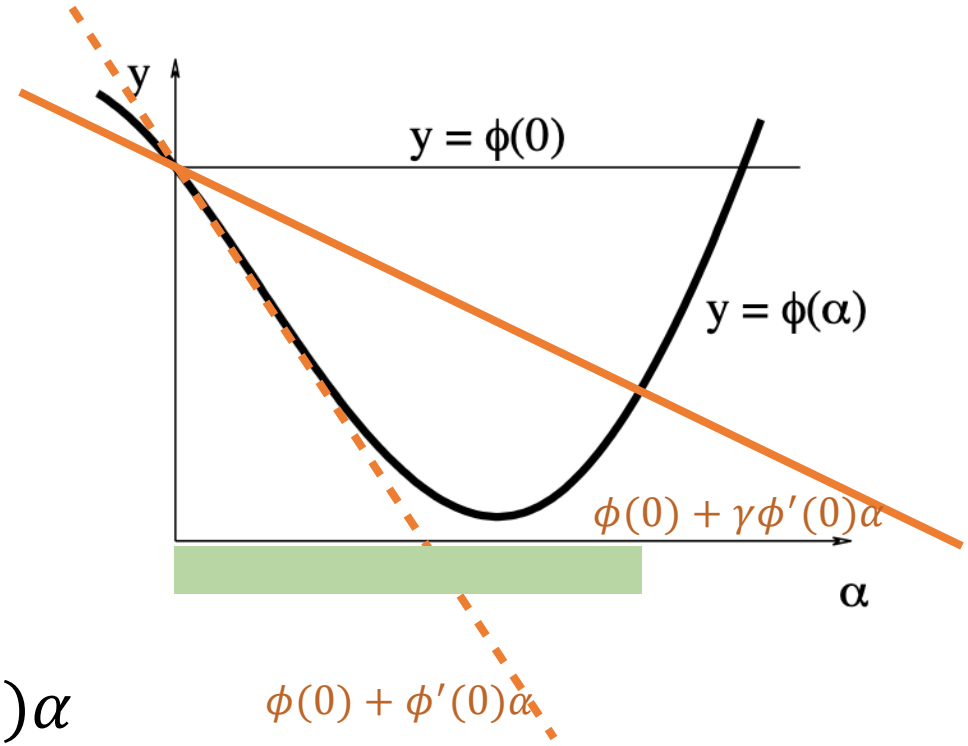
步长求解方法：

1. 穷举（太耗时）

2. 回溯法

- 初始化 α = 较大的值
- 减小 α 直到

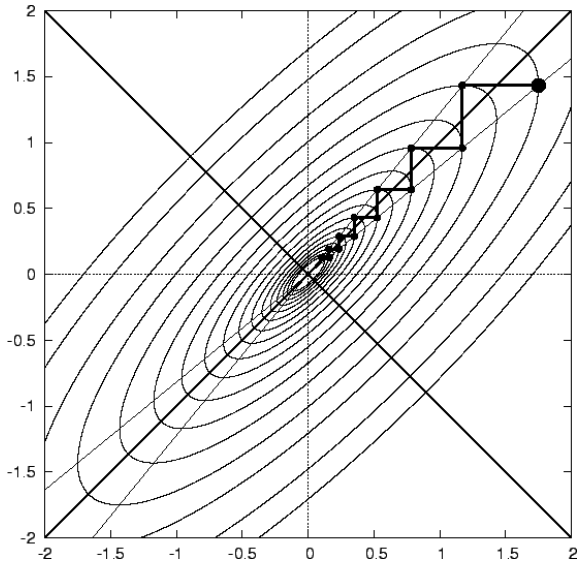
$$\phi(\alpha) \leq \phi(0) + \gamma\phi'(0)\alpha$$



物理含义：满足上式时，α处于右图中绿色区间

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
  - 为什么收敛慢？
    - 只利用了一阶梯度信息
    - 没有利用曲率信息



共轭梯度法

- 给定一个二次型

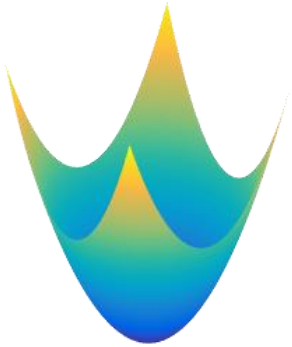
$$f(x) = \frac{1}{2}x^T Ax + bx$$

对称

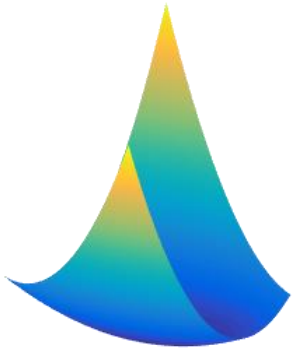
- **它的形状是什么样的？**

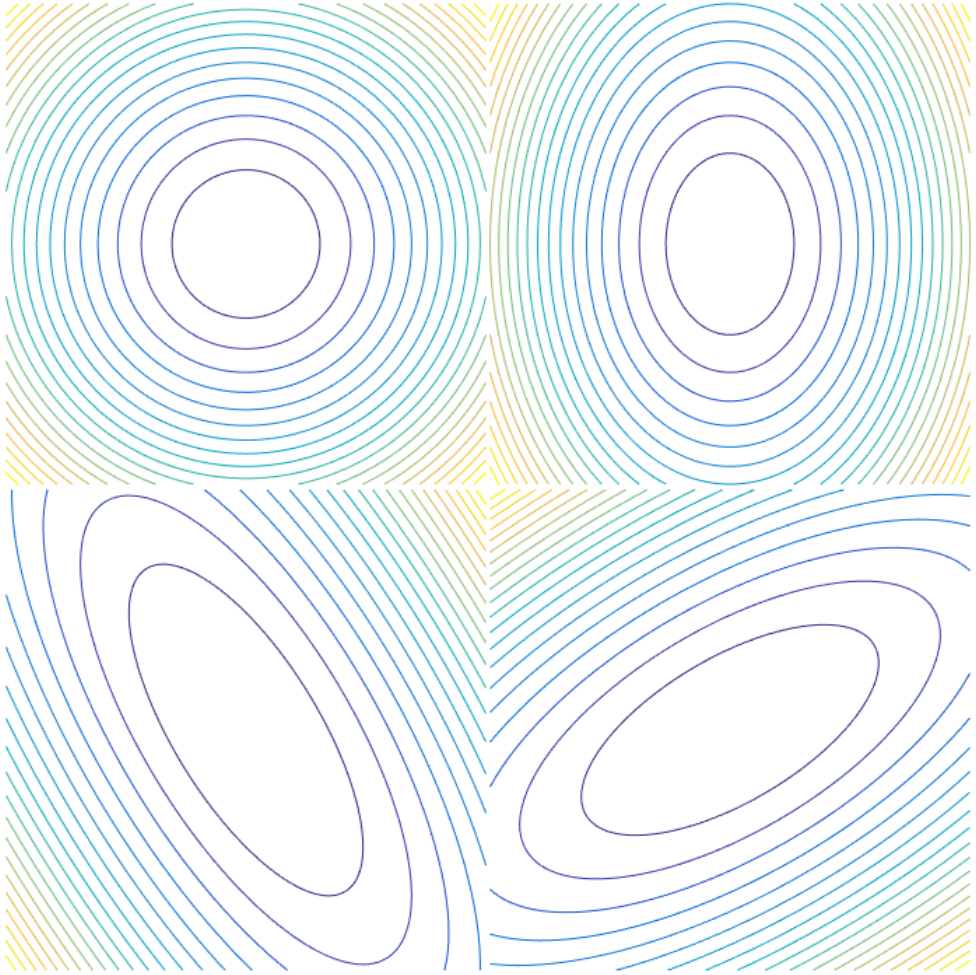
共軛梯度法

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

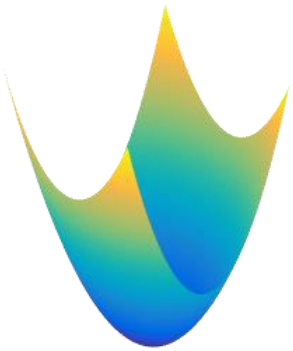


$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$





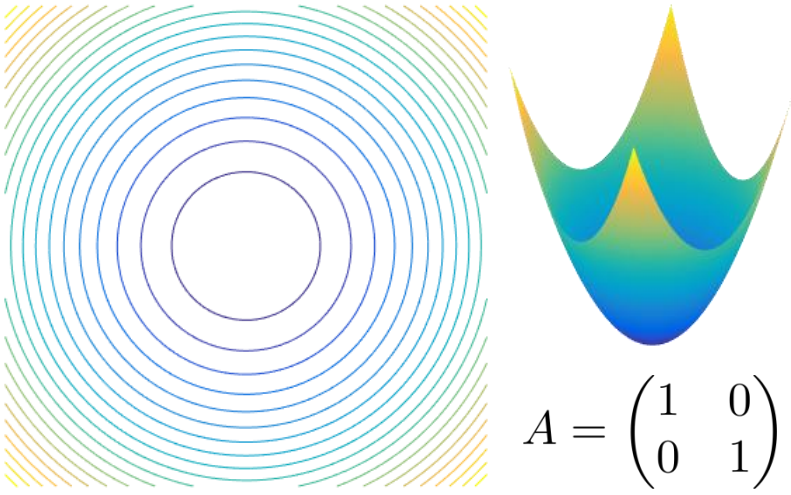
$A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$



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

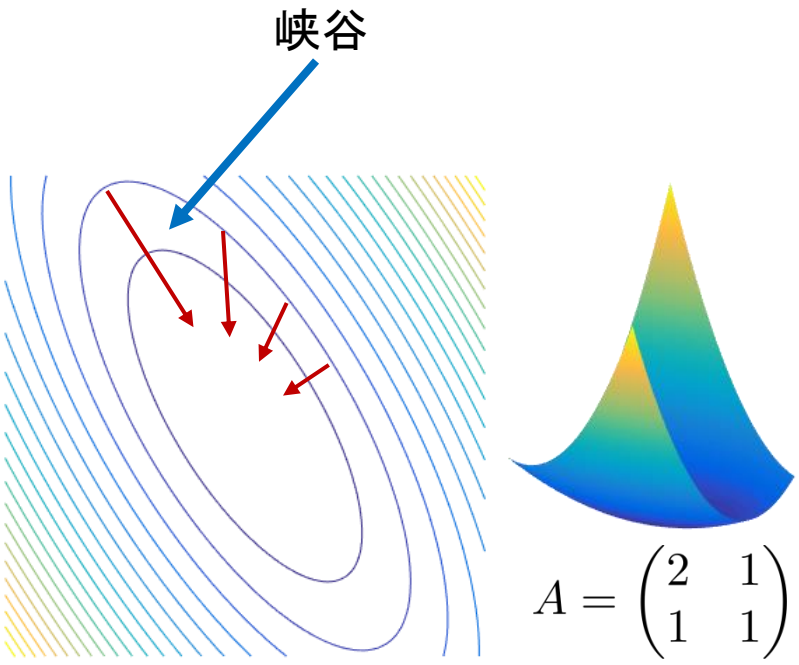
共轭梯度法

- 如果局部很均匀，也就不存在峡谷和峭壁



共轭梯度法

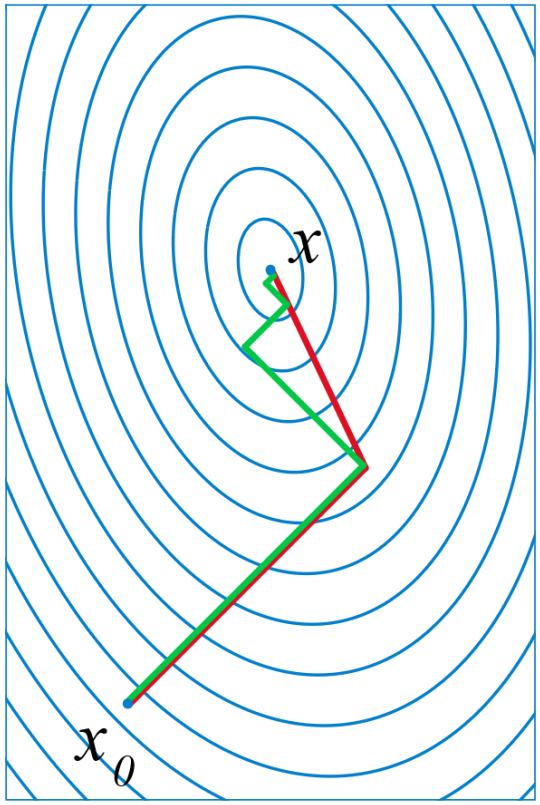
- A包含了局部的曲率信息
  我们可以知道哪里有峡谷



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

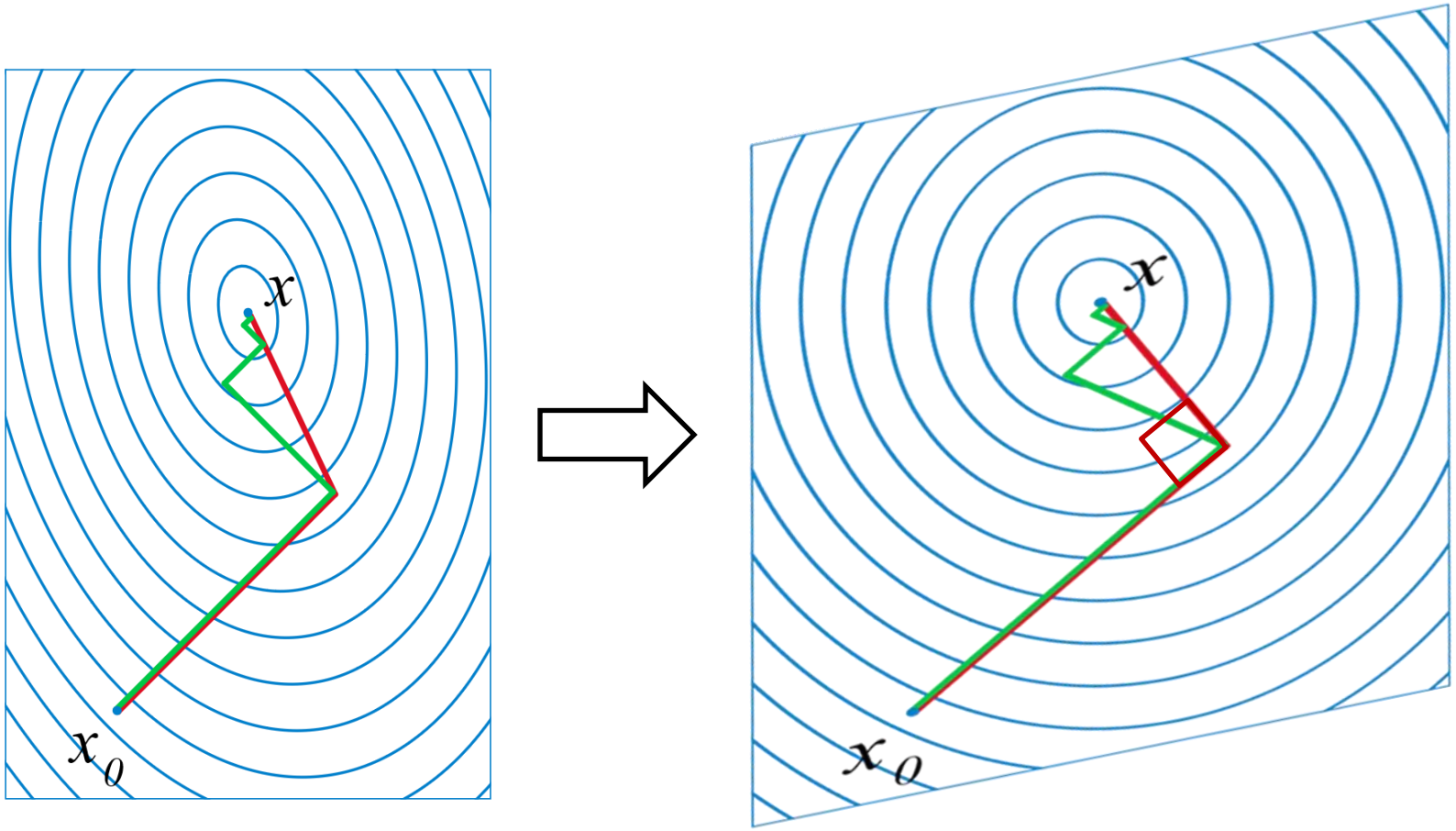


共轭梯度法

- 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A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A

共轭梯度法（非线性）

- $x \leftarrow x_0$
- $p \leftarrow -J_F^T(x)$ // 初始方向采用负梯度方向
- while not converge
  - $x \leftarrow x + \alpha p$
  - $p \leftarrow -J_F^T(x) - \frac{\langle -J_F^T(x), p \rangle_{H_F}}{\langle p, p \rangle_{H_F}} p$

用函数的Hessian诱导共轭

共轭梯度法

- 更新方向的步骤可以简写成
  - $p \leftarrow -J_F^T + \beta p$

- β的选择
  - Fletcher-Reeves/Polak-Ribiere/…
  - $\beta = 0$ ：从目前位置开始新的共轭梯度迭代

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

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 →

牛顿法

- 利用目标函数的二阶导数（曲率）信息
- 在x_k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$F(x_k + \Delta x)$的Δx

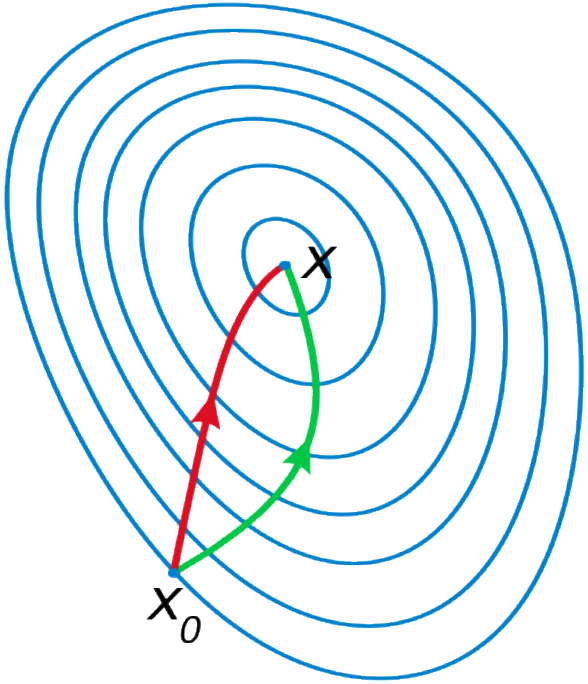
$$H_F \Delta x + J_F^T = 0$$

- 最佳下降方向（牛顿步）：

$$\Delta x = -H_F^{-1} J_F^T$$

牛顿法

- 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?



绿色：梯度下降
红色：牛顿法

高斯牛顿法

- 利用问题的性质来近似求解二阶导
  - 最小二乘问题

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

  - 对R进行一阶近似：

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

  - 此时最优的Δx满足

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

  - 最优下降方向：

$$\Delta x = -(J_R^T J_R)^{-1} J_R^T R(x_k)$$

高斯牛顿法

- 最优下降方向：

$$\Delta x = -(J_R^T J_R)^{-1} J_R^T R(x_k)$$

$$= -(J_R^T J_R)^{-1} J_F^T$$

- 对比牛顿法：
  - 牛顿步：$\Delta x = -H_F^{-1} J_F^T$
  - 高斯牛顿法用$J_R^T J_R$近似Hessian矩阵H_F

- 优点
  - 不需要Hessian，容易计算
  - 收敛快

- 缺点
  - 如果$J_R^T J_R$不可逆，算法会变得不稳定

Levenberg-Marquardt

- LM法通过“正则化”回避这个问题

$$\Delta x = -(J_R^T J_R + \lambda I)^{-1} J_R^T R(x_k)$$

- 对于全部$\lambda > 0$，$J_R^T J_R + \lambda I$一定是正定的

Levenberg-Marquardt

$$\Delta x = -(J_R^T J_R + \lambda I)^{-1} J_R^T R(x_k)$$

- λ的效果
  - $\lambda \to \infty$： 梯度下降步，并且长度短
  - $\lambda \to 0$： Gauss-Newton 步

- λ的选择
  - 每轮迭代更新
  - 当下降明显时，$\lambda \downarrow$
  - 当下降不明显时，$\lambda \uparrow$

Levenberg-Marquardt

- 优点:
  - 启动快 ($\lambda \uparrow$)
  - 收敛快 ($\lambda \downarrow$)
  - 不退化 ($J_R^T J_R + \lambda I$总是正定)
  - LM = 梯度下降+高斯牛顿

Constrained optimzation

- 需要考虑目标函数与约束的形式

Constrained optimization

In mathematical optimization, **constrained optimization** (in some contexts called **constraint optimization**) is the process of optimizing an objective function with respect to some variables in the presence of constraints on those variables. The objective function is either a cost function or energy function, which is to be minimized, or a reward function or utility function, which is to be maximized. Constraints can be either **hard constraints**, which set conditions for the variables that are required to be satisfied, or **soft constraints**, which have some variable values that are penalized in the objective function if, and based on the extent that, the conditions on the variables are not satisfied.

Contents

Relation to constraint-satisfaction problems

General form

Solution methods

- Equality constraints
  - Substitution method
  - Lagrange multiplier
- Inequality constraints
  - Linear programming
  - Nonlinear programming
  - Quadratic programming
  - KKT conditions
  - Branch and bound
  - First-choice bounding functions
    - Russian doll search
  - Bucket elimination

See also

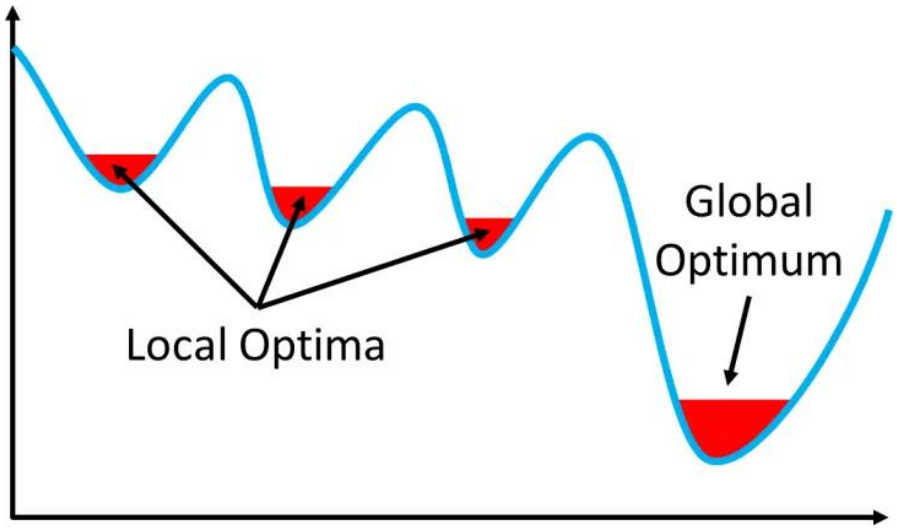
References

Further reading

https://en.wikipedia.org/wiki/Constrained_optimization

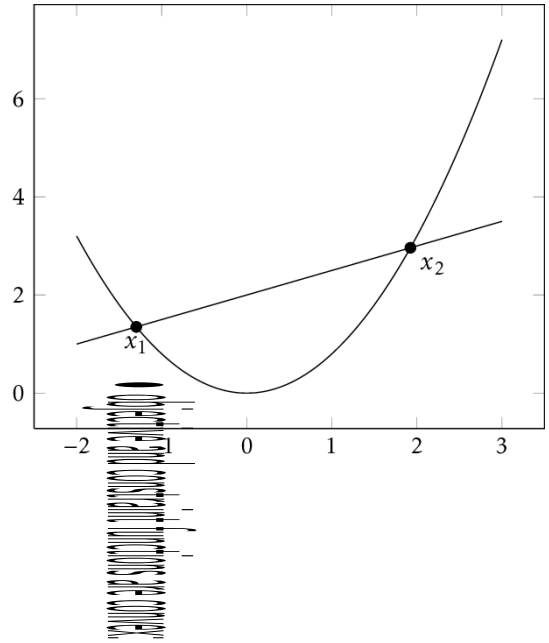
局部最优与全局最优

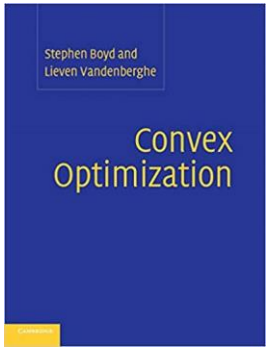
- **梯度下降只能找到局部最**优解
- 对于特殊性质的函数，局部最优即全局最优（比如凸函数）



凸优化

- Convex optimization





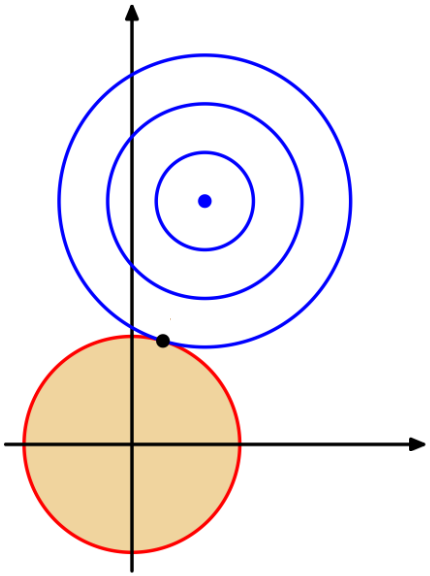
https://web.stanford.edu/class/ee364a/

病态问题

- 病态问题指的是问题的解不唯一
  例如：$Ax = b$ 当方程数小于变量数

- 正则化：通过添加先验知识来约束解的性质

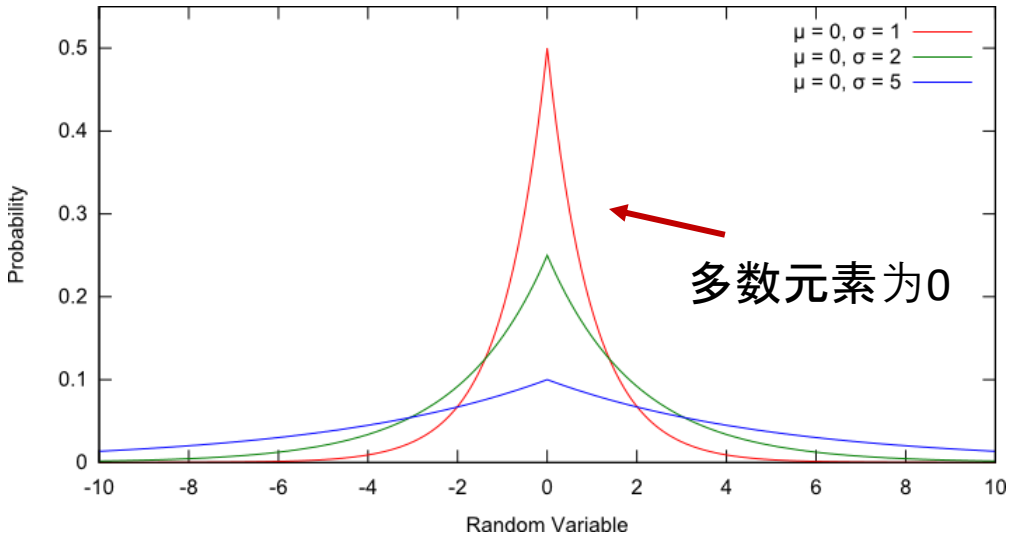
- L2正则：$\min_x \|Ax - b\|_2^2 + \lambda\|x\|_2^2$
  - L2正则可以让x的值趋向于0
  - 作用：抑制冗余的变量



正则化

- 一些正则化项对应了模型的先验概率
  - 假设x的每个系数x_i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：x的参数符合Laplace分布

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

  - 已知A, b时x的后验概率是

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

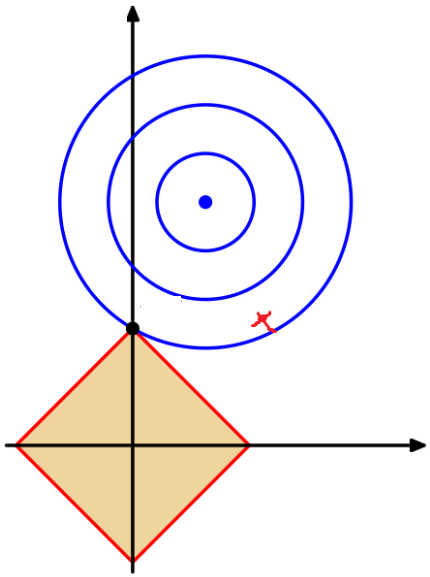
正则化

- 在已知A,b条件下的最有可能的x满足最大后验概率估计：

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

L1正则

- L1范数：$\| x \|_1 = \sum_i |x_i|$
- L1正则：$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_1$
- L1正则可以让x变得稀疏



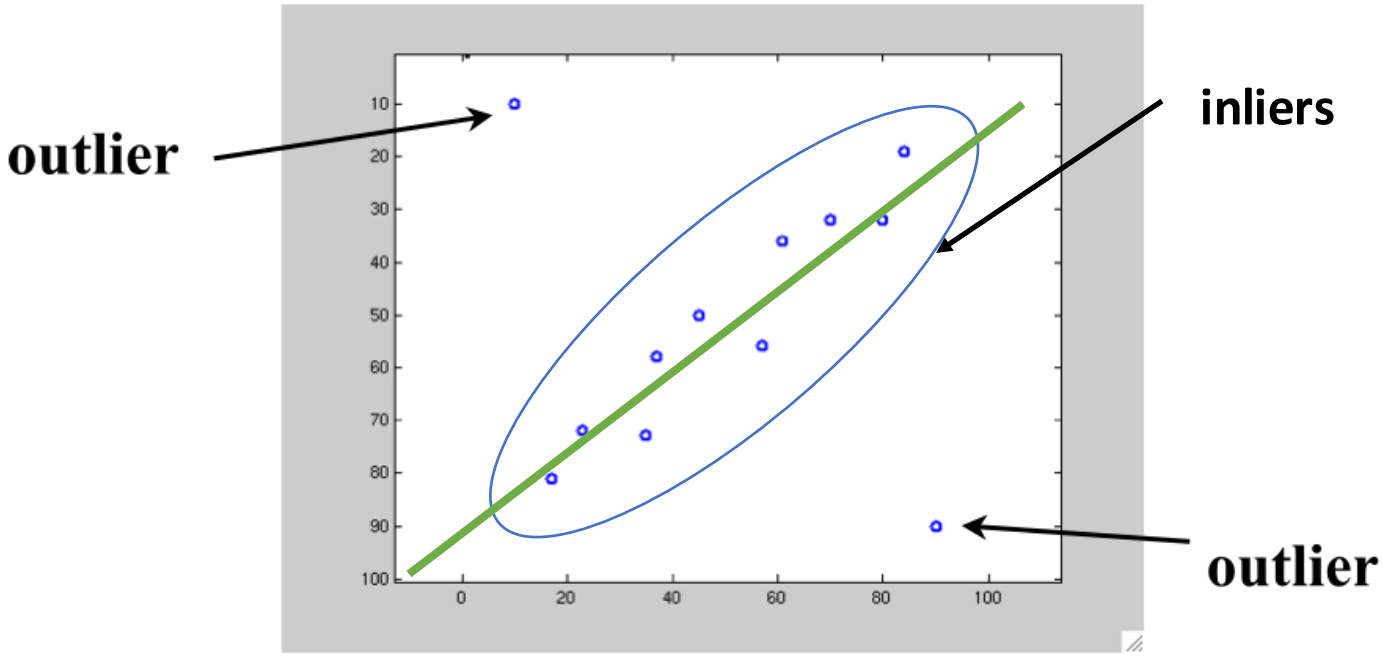
正则化

- 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

正常值与离群值

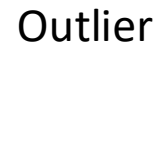
- 正常值（Inlier）：满足模型/噪音假设的数据点
- 离群值（Outlier）：不满足模型/噪音假设的错误点



离群值

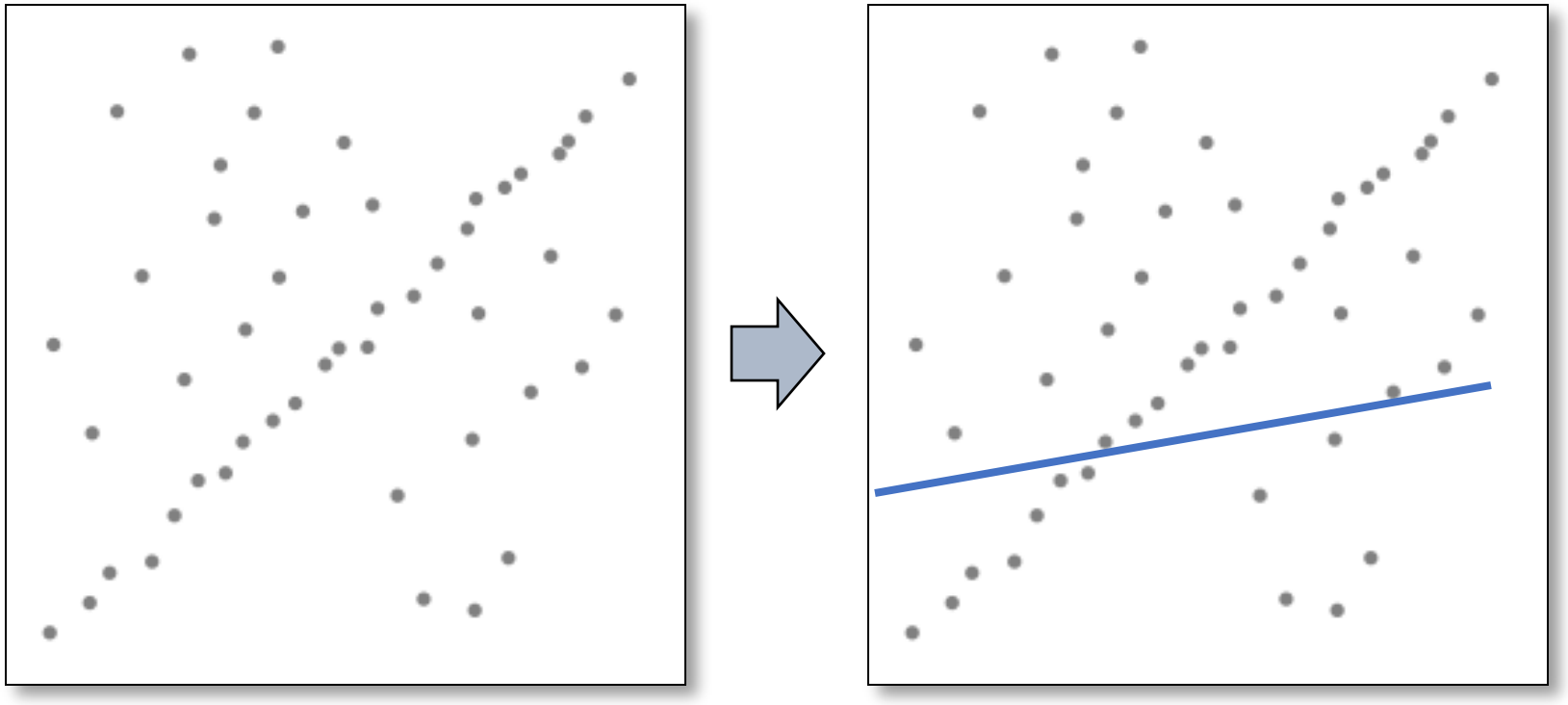
- 离群值 = 不满足模型/噪音假设的错误点

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



离群值

- 离群值会让最小二乘回归失败

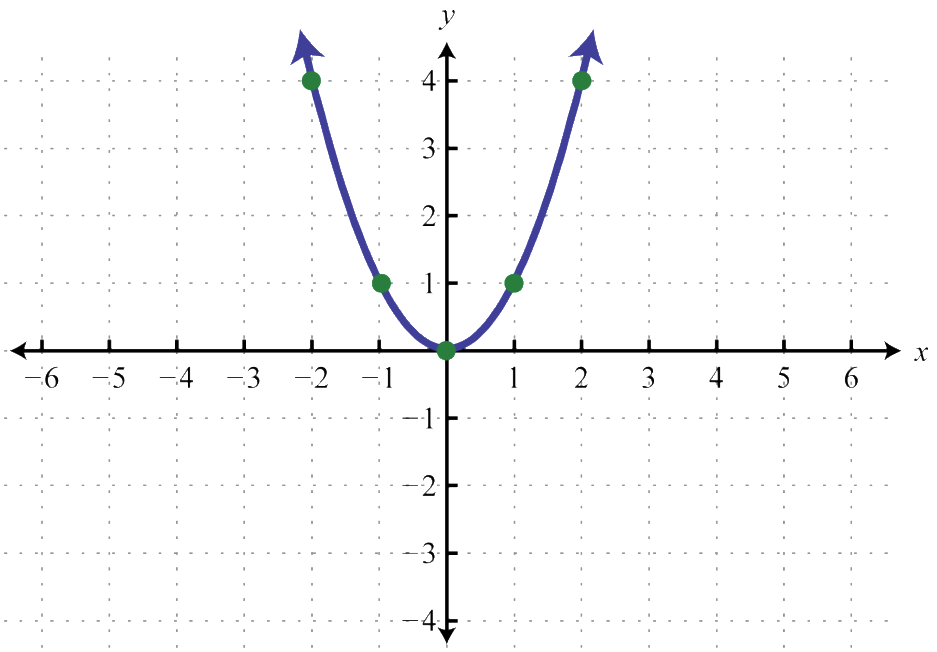


离群值

- 为什么离群值会让最小二乘回归失败
  - MSE（平方误差）与残差平方成正比
  - 受离群值影响过大

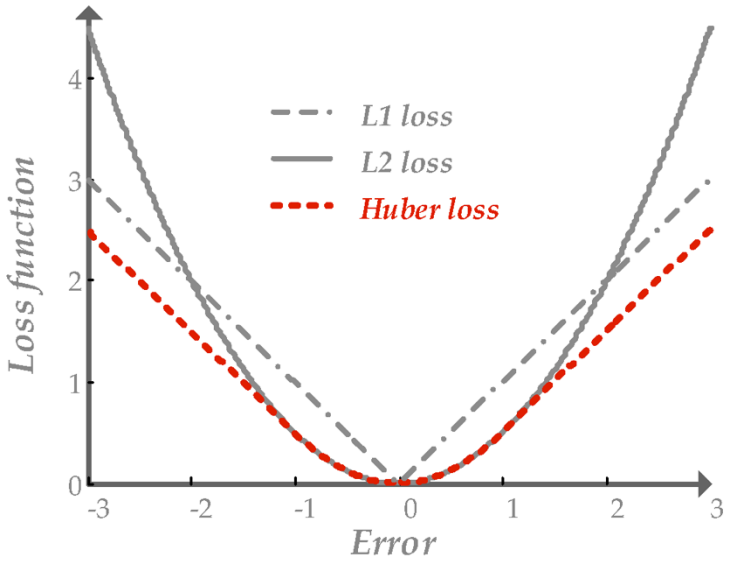
$$f(x) = x^2$$

x	$f(x)$
-2	4
-1	1
0	0
1	1
2	4



鲁棒估计

- 如何降低离群值的影响?
  - 用其他目标函数替换MSE，降低对离群值的惩罚
  - 例如L1范数、Huber函数



  - 这种目标函数一般称为鲁棒函数

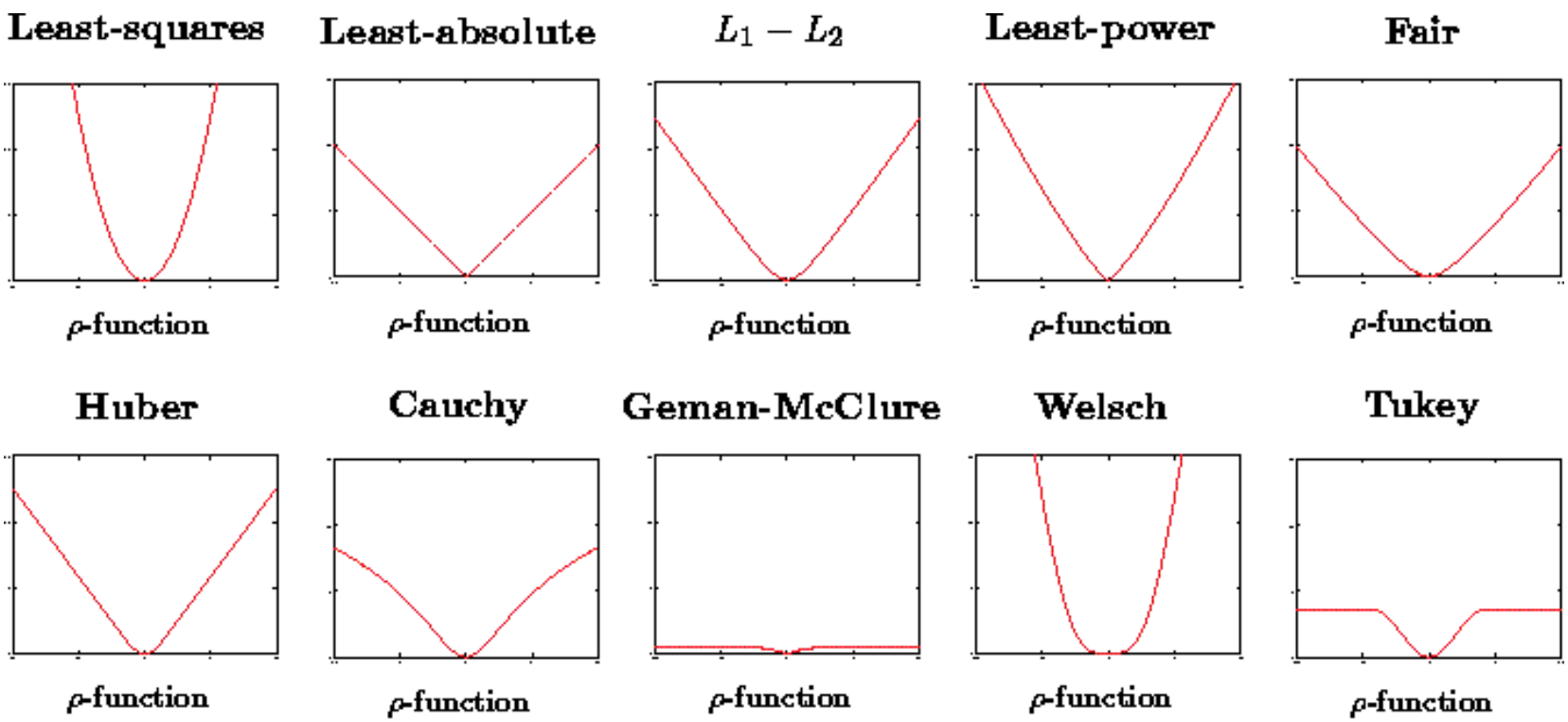
鲁棒最小二乘

- 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

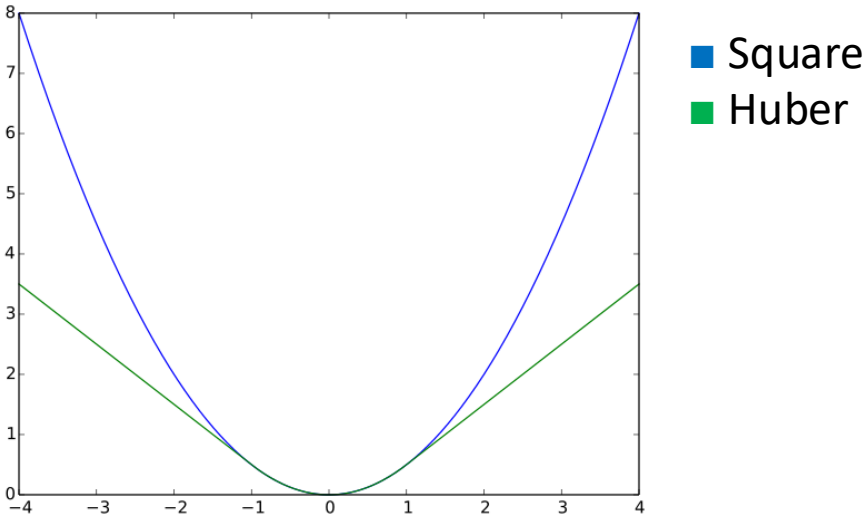
- 引入鲁棒函数ρ(residual)代替$\|\text{residual}\|_2$
  - $\rho(-x) = \rho(x)$
  - $\arg\min_x \rho(x) = 0$
  - $\rho'(x) = 0 \Leftrightarrow x = 0$
  - $O\big(\rho(x)\big) \leq O(\|x\|_2)$

M估计


Least-squares
Least-absolute
$L_1 - L_2$
Least-power
Fair
ρ-function
ρ-function
ρ-function
ρ-function
ρ-function
Huber
Cauchy
Geman-McClure
Welsch
Tukey
ρ-function
ρ-function
ρ-function
ρ-function
ρ-function

M估计

- Huber function：大误差使用线性距离



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

鲁棒估计

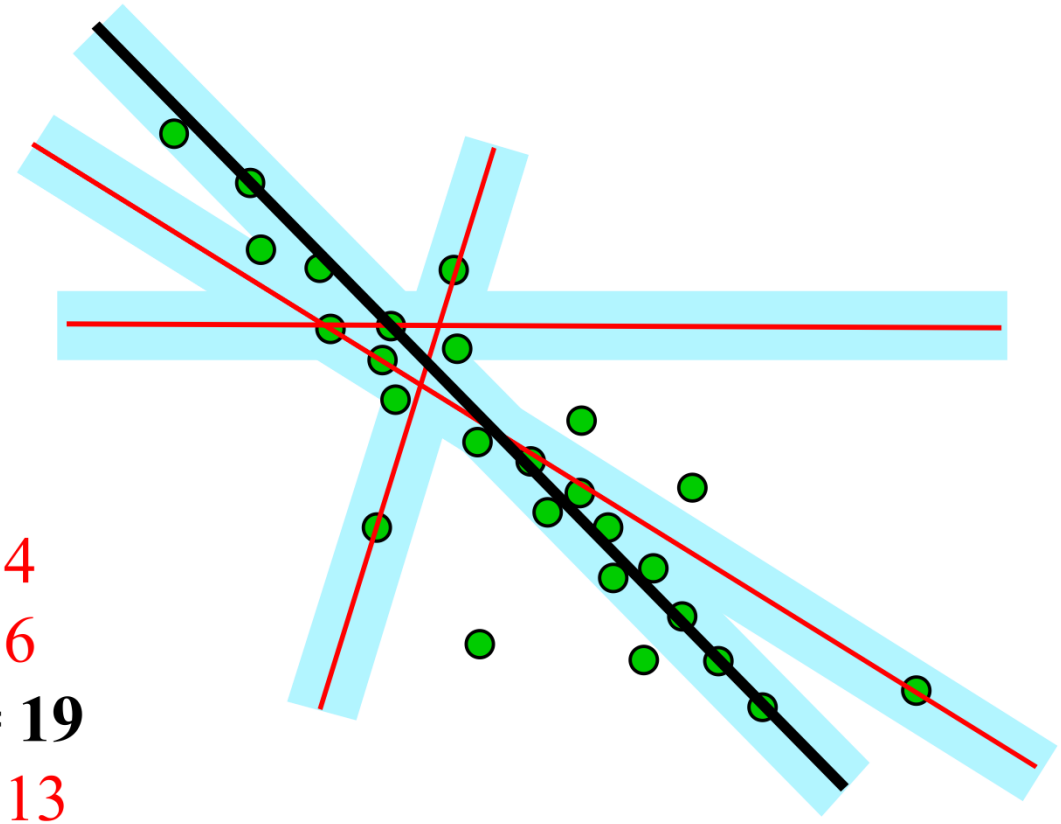
- 除了M估计，还有各种鲁棒过程
  - RANSAC

RANSAC

- 除了鲁棒函数，还有其他处理离群值的方法
- RANSAC: Random Sample Concensus
  - 目前最常用的处理离群值的方法
  - 核心思想：
    - Inlier的分布总是相似，outlier的分布各有各的不同
    - 用数据点对可能的模型参数进行投票

Robert Collins
CSE486, Penn State

Ransac Procedure



Count = 4

Count = 6

Count = 19

Count = 13