数学优化

1、最小二乘least-squares

优化

变形

线性规划

凸优化

非线性优化

数学优化

- x：优化变量，未必唯一解
- 在满足条件的情况下，最小化优化变量目标函数f0：R^n-->R
- x*：最优解，且满足m个约束条件

- [ ] 三类问题易于解决

```
最小二乘
线性规划
凸优化
```

1、最小二乘least-squares

> 无约束条件
> 平方只是拟合手段
> 二范数平方=平方和，易于识别
> 可以加权weighed

$$\min \quad \|Ax-b\|_2^2=\sum_{i=1}^{k}\left(a_i^{\mathsf{T}}x-b_i\right)^2$$

> 向量默认都是列向量
> 思维需要从标量转到向量

优化

> 无约束条件，自然想到求导算最优
> 求解线性方程得到最小二乘优化问题的最优解
> 软件成熟高效，复杂度k*n^2（为大规模问题可以设计优化算法）

$$
\begin{aligned}
& 2A^\top(Ax-b)=0 \\
\Longrightarrow \quad & A^\top Ax = A^\top b \\
\Longrightarrow \quad & x=(A^\top A)^{-1}A^\top b
\end{aligned}
$$

正则化，平方和仍是最小二乘，

变形

> 实际都还是最小二乘

```
加权weighed
正则化regularization
```

> 避免过拟合，可逆，更稳定

- Weighted least-squares

$$\sum_{i=1}^{k} w_i\left(a_i^\top x - b_i\right)^2 = \sum_{i=1}^{k}\left(\sqrt{w_i}a_i^\top x - \sqrt{w_i}b_i\right)^2$$

  - Different importance

- Regularization

$$\sum_{i=1}^{k}\left(a_i^\top x - b_i\right)^2 + \rho\sum_{i=1}^{n} x_i^2$$

  - More stable

线性规划

> 当目标函数和约束函数都线性时
> 有约束不能求导=0
> 不易识别

- 线性函数：加权的函数等于函数的加权
- 时间复杂度n^2*m，m约束条件个数，n优化变量维度

Chebyshev Approximation Problem

$$
\begin{aligned}
& \min \quad \max_{i=1,\ldots,k} \left|a_i^\top x - b_i\right| \\
\Leftrightarrow \quad & \min \quad t \\
& \text{s.t.} \quad t = \max_{i=1,\ldots,k} \left|a_i^\top x - b_i\right| \\
\Leftrightarrow \quad & \min \quad t \\
& \text{s.t.} \quad t \geq \left|a_i^\top x - b_i\right|, i = 1, \ldots, k \\
\Leftrightarrow \quad & \min \quad t \\
& \text{s.t.} \quad -t \leq a_i^\top x - b_i \leq t, i = 1, \ldots, k
\end{aligned}
$$

> 多了一个t，但变成了线性规划问题

凸优化

> 最小二乘和线性规划都是特例：=变<=，且α+β=1并且为正
> 其实要求放松了，不需要对所有α,β都成立

- 局部最优一定是全局最优
- 识别困难

> 看起来不像，但能变化成凸优化

- 时间复杂度max{n^3,m*n^2,F}

非线性优化

> 无高效优化，复杂度高

- 找局部最优解

> 神经网络，局部解已经够用
> 需要给一个初始解，可能需要凸优化方法给出
> 是艺术而不是技术

- 和凸优化区别

> 问题本身形式化出来容易，只要能求导就行，但求解困难

> 问题想转化成凸优化很难，靠技巧和灵感。只要建模成凸优化，求解容易

- 全局最优

> 变量少，有超级计算机，和生命安全有关必须全面时，才能做到
> 对最坏情况分析
> 局部优化法可以举反例证明不可靠

- 和凸优化联系

> 凸优化得到非凸优化的下届