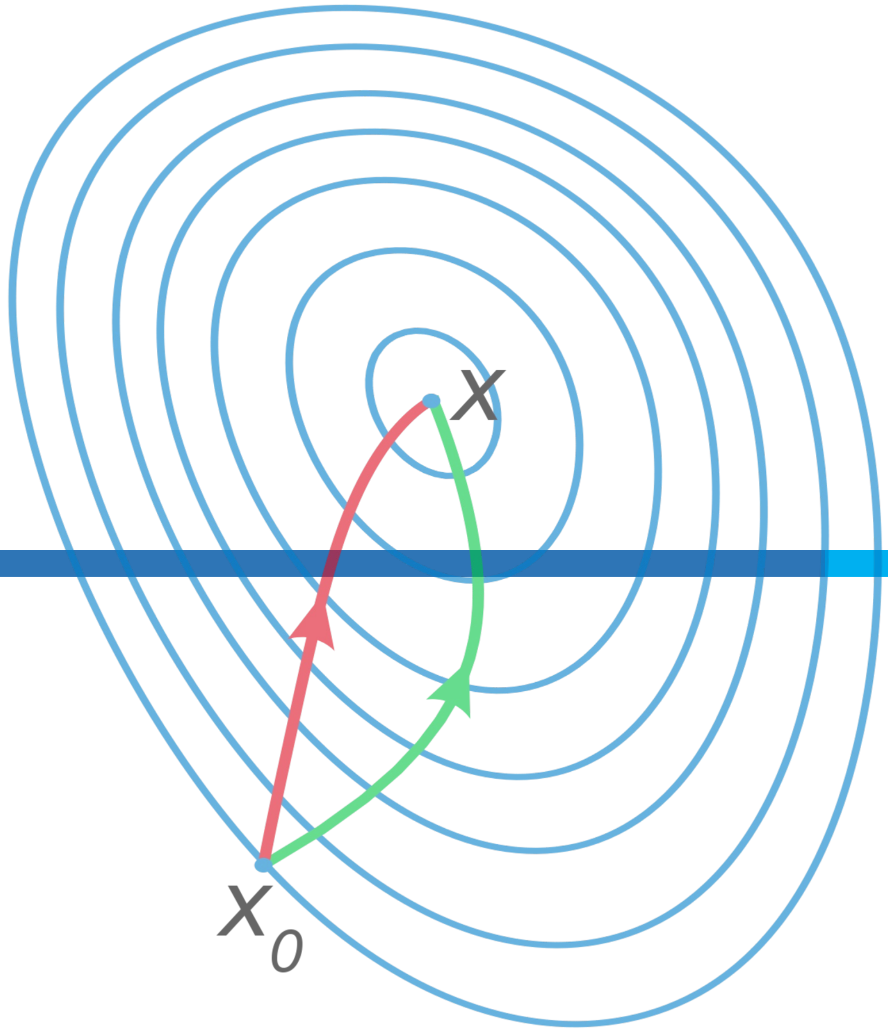
最优化方法

第一周



计算机学院
余皓然
2024/2/26

什么是最优化方法？

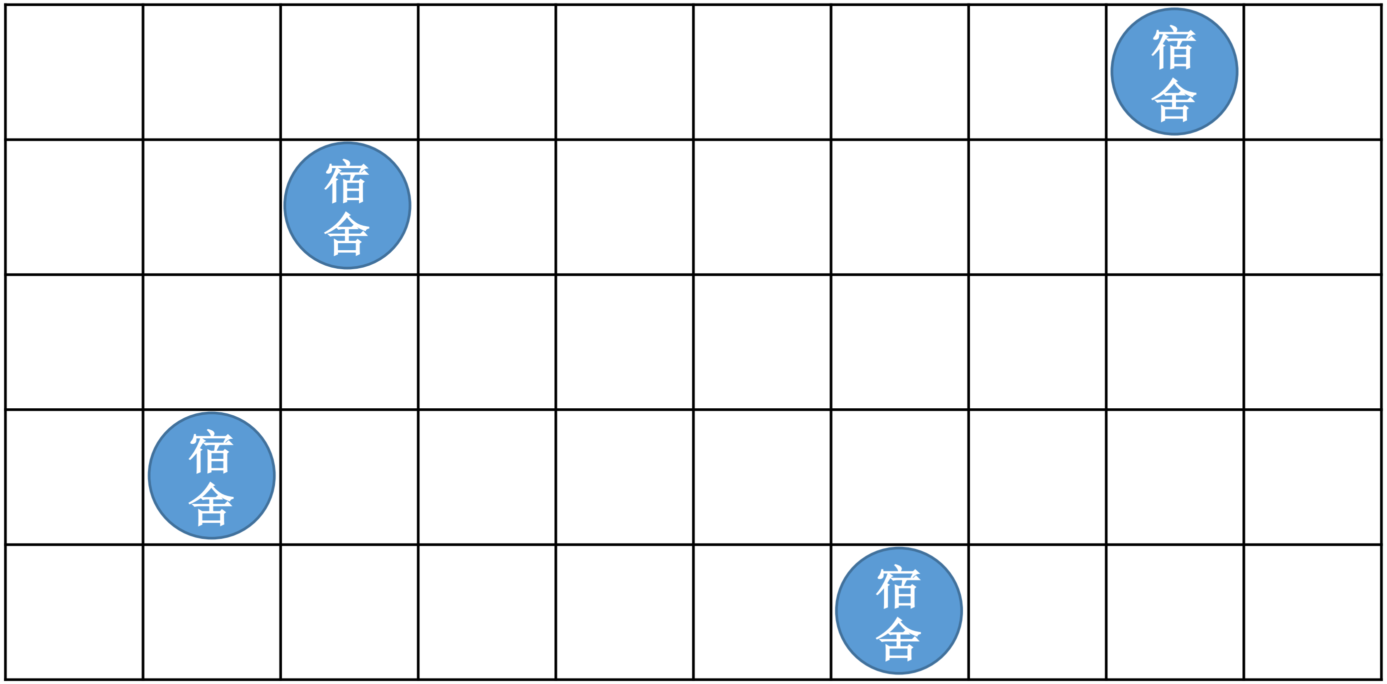
最优化方法

- 最优化方法：解决最优化问题的方法

- 最优化问题（Optimization Problem）：在一定约束条件下，如何选择变量，从而最大化/最小化预设目标

最优化问题示例

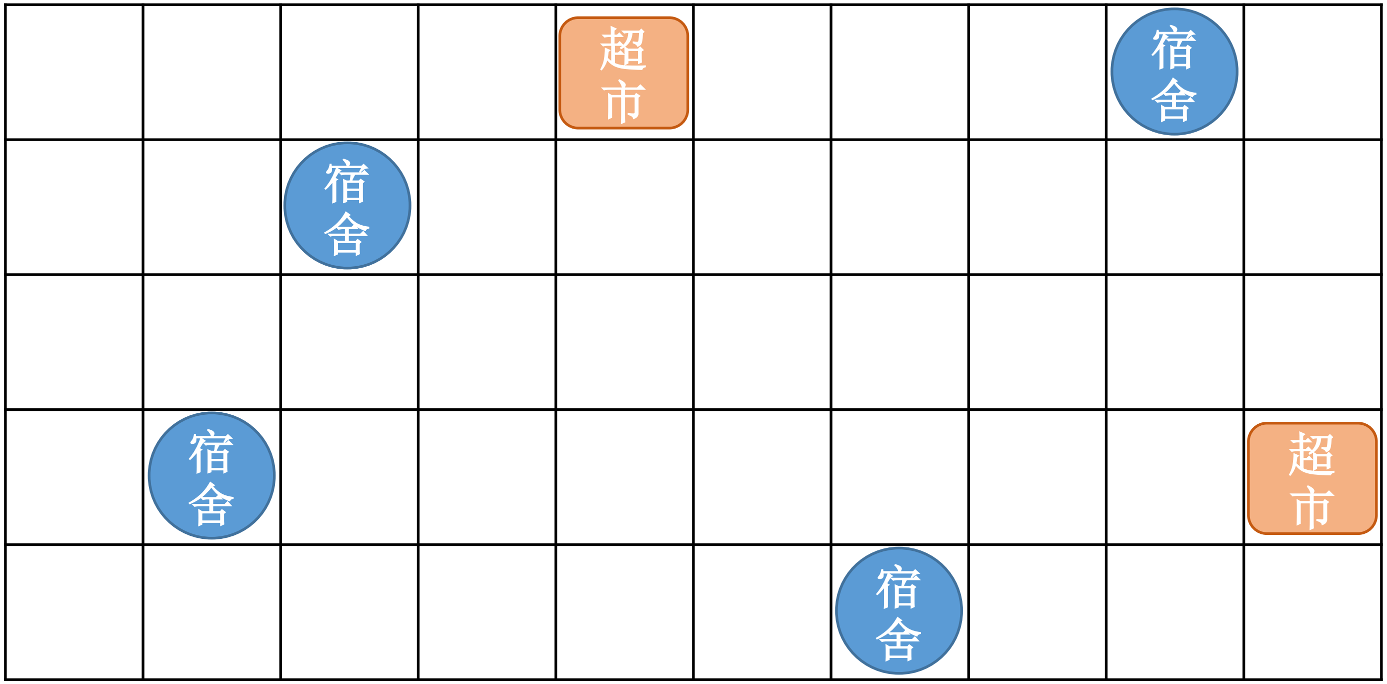
校区要新建两个超市，问如何选址可以最小化每个宿舍到离其最近超市的距离之和?



例子取自Harvard CS50's Introduction to Artificial Intelligence with Python课程

最优化问题示例

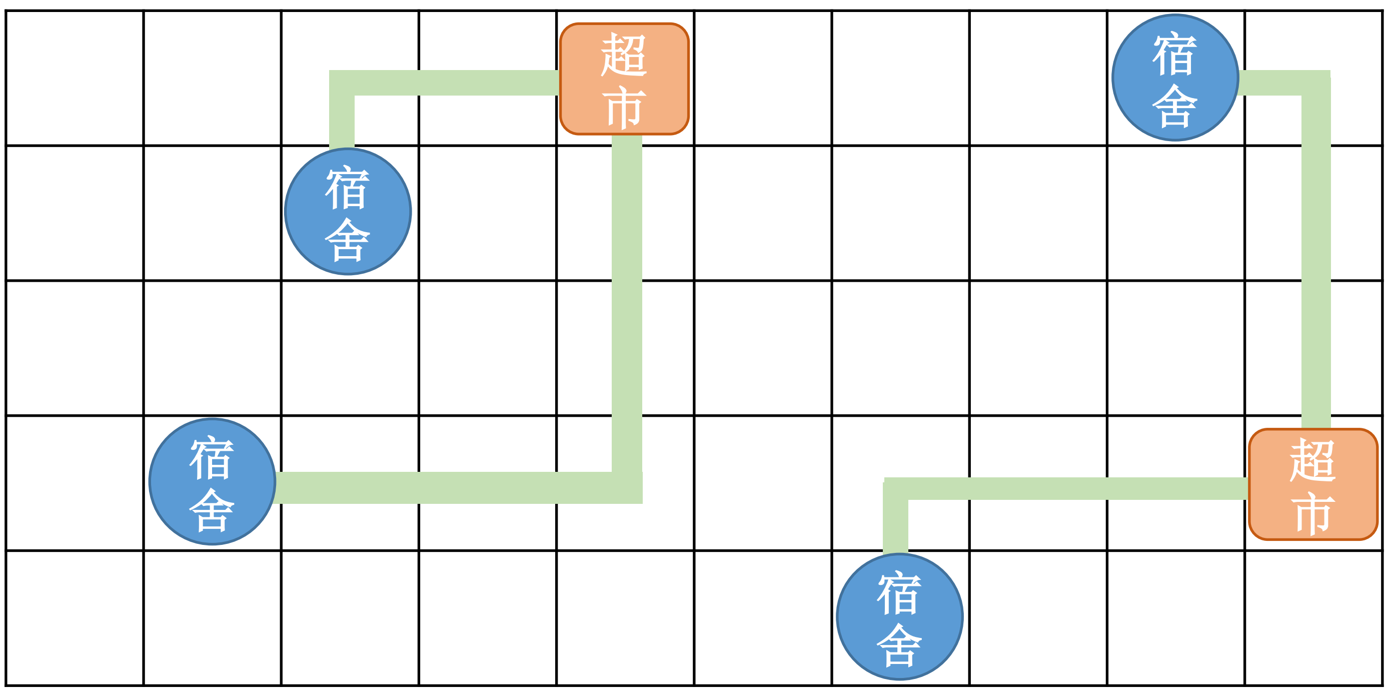
校区要新建两个超市，问如何选址可以最小化每个宿舍到离其最近超市的距离之和?



以上选址方案对应的每个宿舍到离其最近超市距离之和（目标值）为多少?

最优化问题示例

校区要新建两个超市，问如何选址可以最小化每个宿舍到离其最近超市的距离之和?

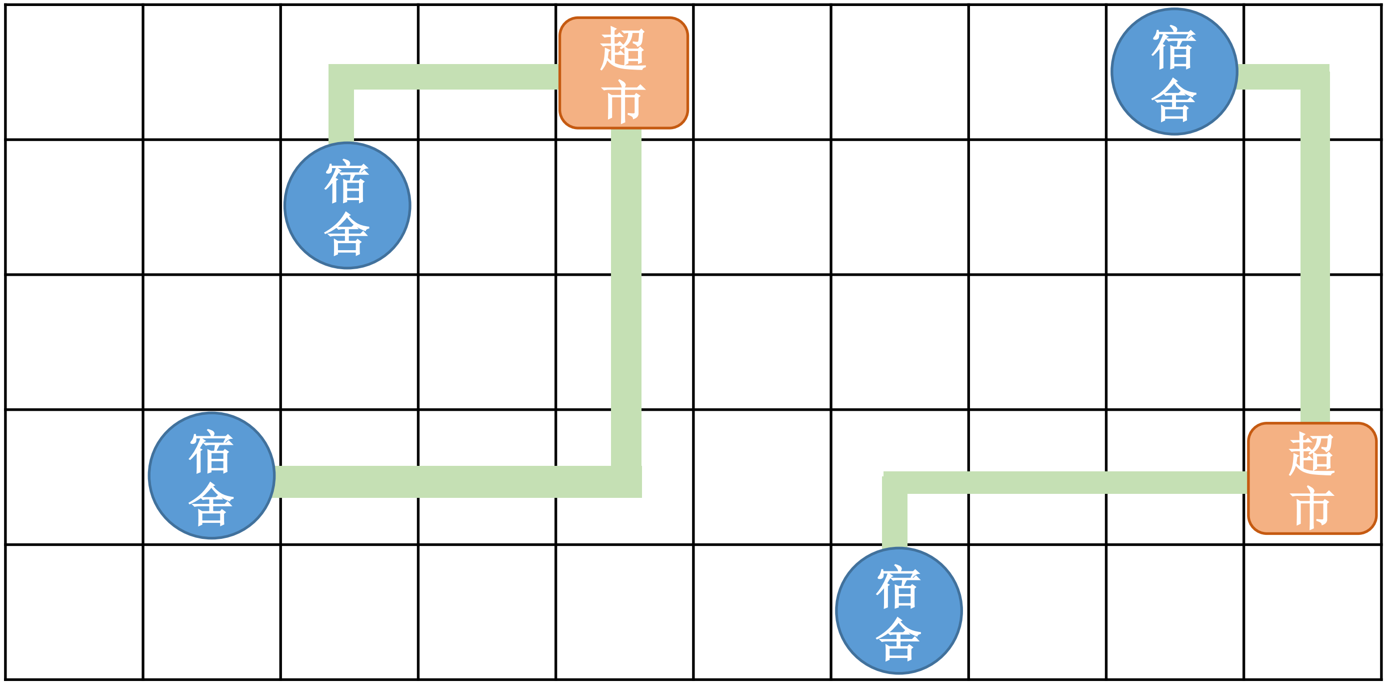


以上选址方案对应的目标值为17

最优化问题示例

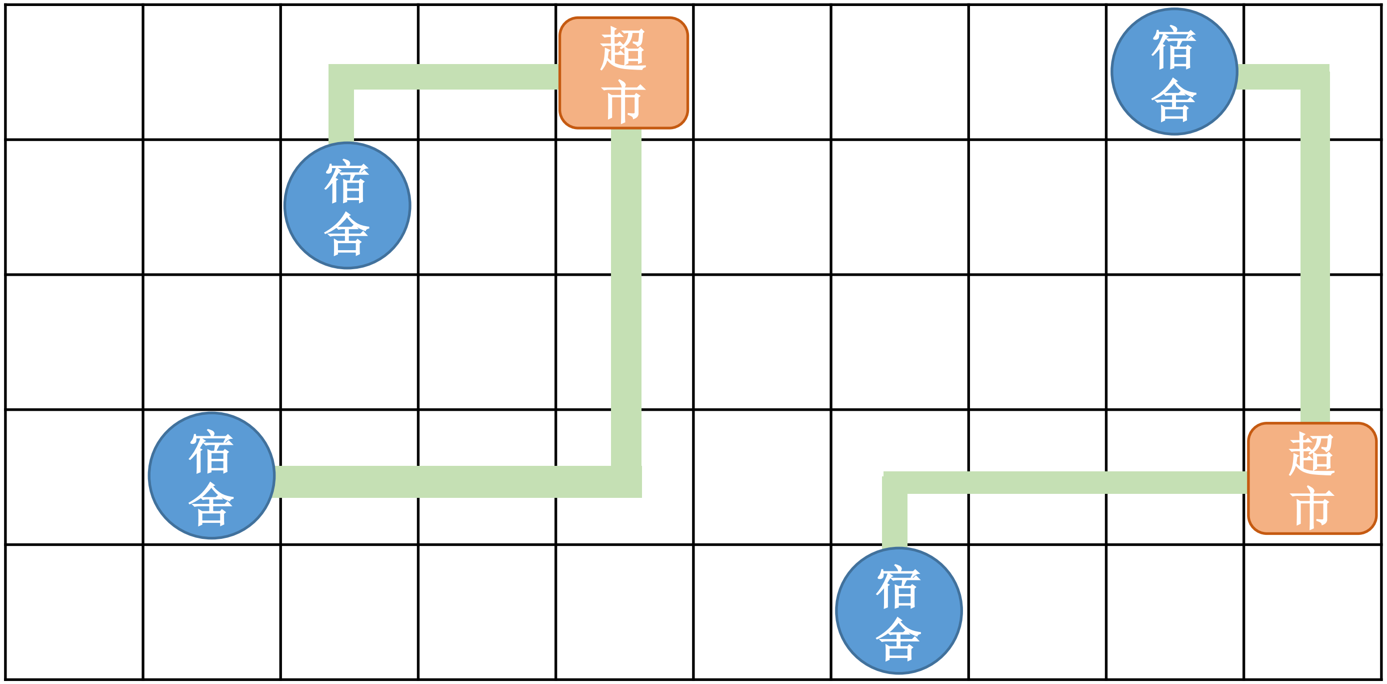
最优化问题：在一定约束条件下，如何选择变量，从而最大化/最小化预设目标

校区要新建两个超市，问如何选址可以最小化每个宿舍到离其最近超市的距离之和?



最优化方法示例

采用什么最优化方法求解?



最优化方法示例

采用什么最优化方法求解?
穷举法 (brute-force search/exhaustive search)?

1	2	3	4	5	6	7	8	宿舍	9
10	11	宿舍	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28
29	宿舍	30	31	32	33	34	35	36	37
38	39	40	41	42	43	宿舍	44	45	46

最优化方法示例

穷举法的缺点是什么？原因是什么？
能不能由此想到其它的方法？

1	2	3	4	5	6	7	8	宿舍	9
10	11	宿舍	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28
29	宿舍	30	31	32	33	34	35	36	37
38	39	40	41	42	43	宿舍	44	45	46

最优化方法示例

登山搜索：**根据一定的方向**在解空间进行搜索

1	2	3	4	5	6	7	8	宿舍	9
10	11	宿舍	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28
29	宿舍	30	31	32	33	34	35	36	37
38	39	40	41	42	43	宿舍	44	45	46

最优化方法示例

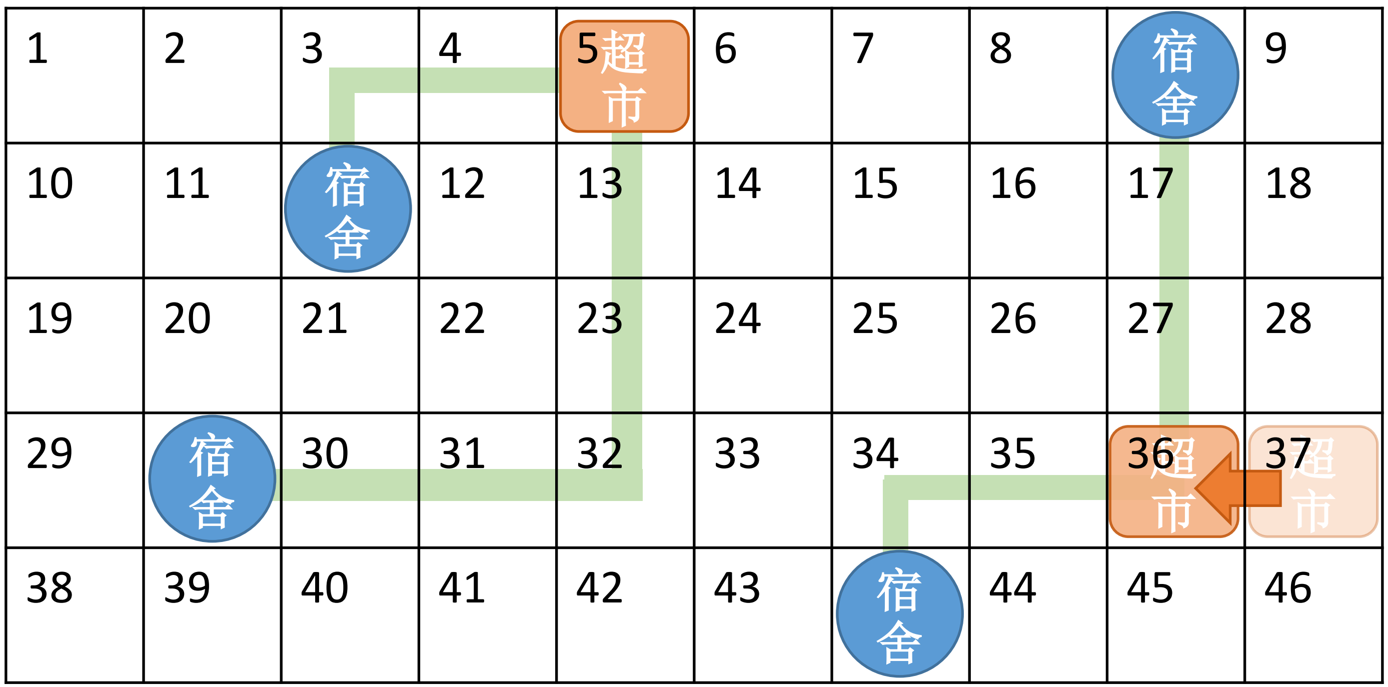
针对现有方案有很多种可能的更改方案，假设仅考虑将一个超市移动一格的更改方案，共6种该类更改方案

1	2	3	4 超市	5 超市	6 超市	7	8	宿舍	9
10	11	宿舍	12	13 超市	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28 超市
29	宿舍	30	31	32	33	34	35	36 超市	37 超市
38	39	40	41	42	43	宿舍	44	45	46 超市

以上选址方案对应的目标函数值为17

最优化方法示例

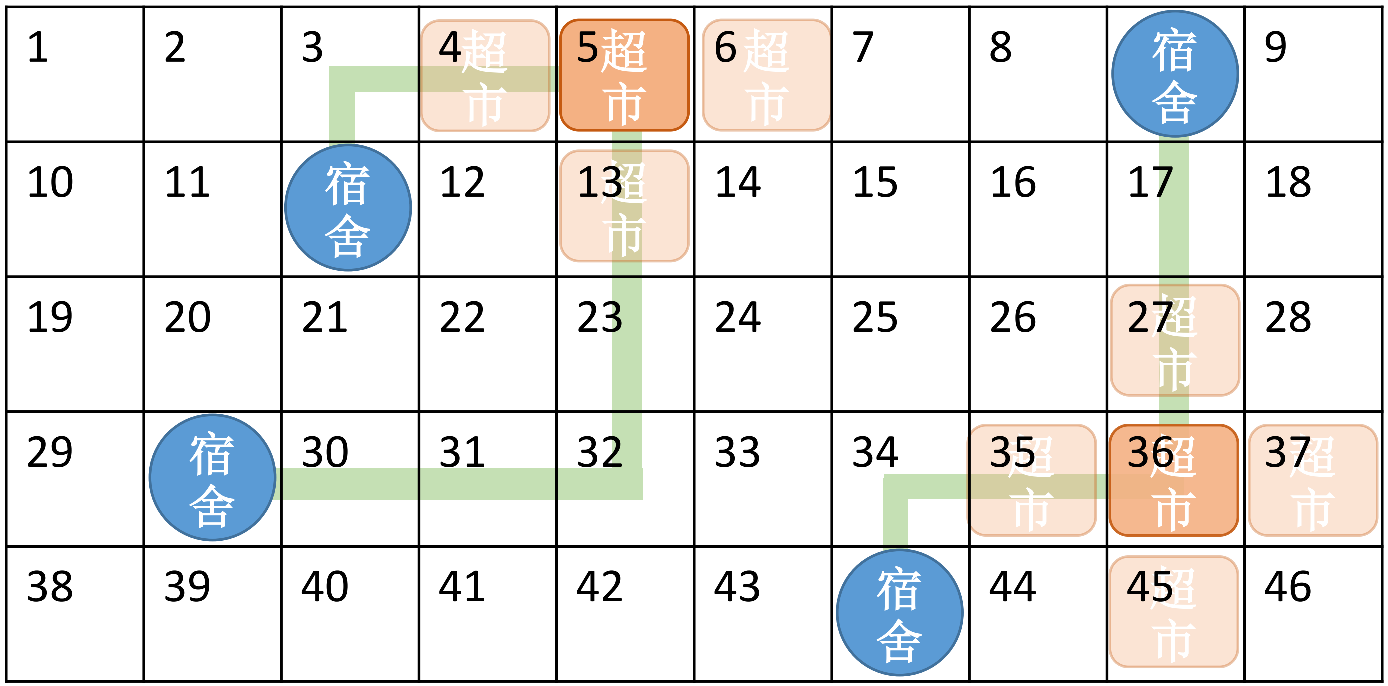
从6种该类更改方案中选择对应目标函数最低的方案



选址方案对应的目标函数值从17降低为15

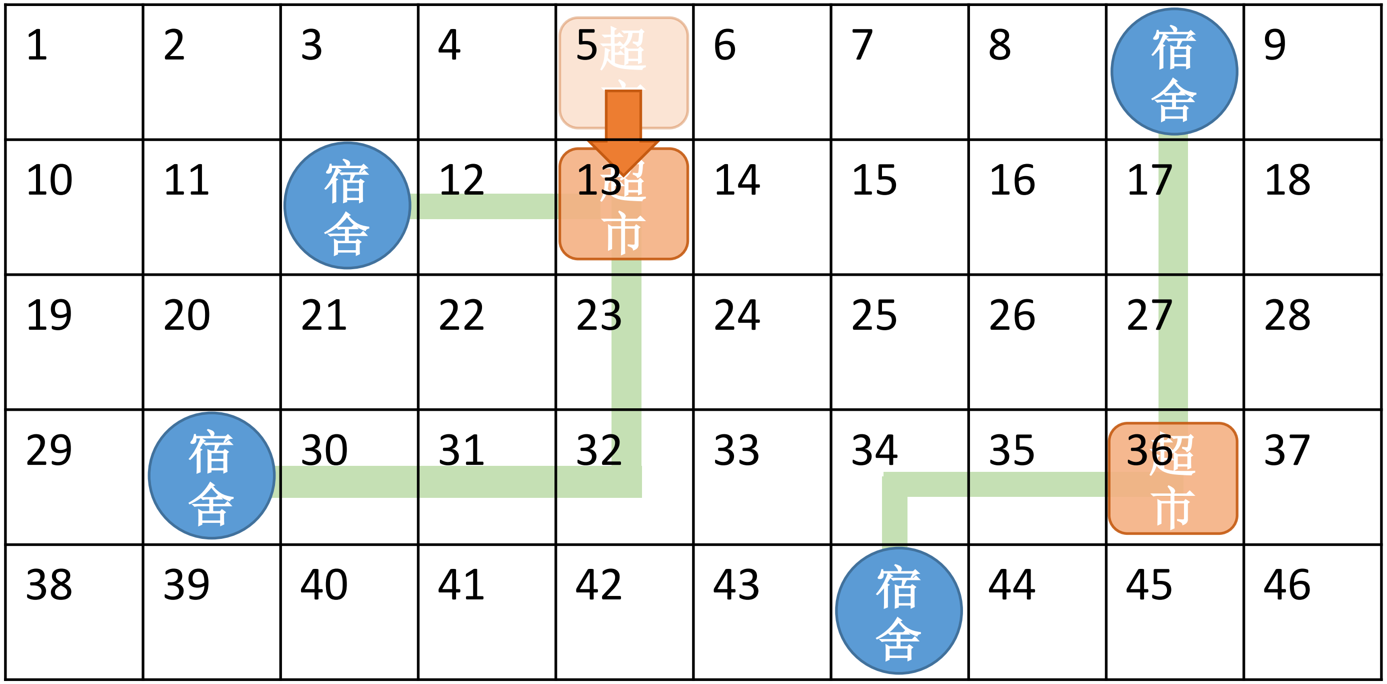
最优化方法示例

针对新方案又有7种更改方案
（仍仅考虑将一个超市移动一格的更改方案）



最优化方法示例

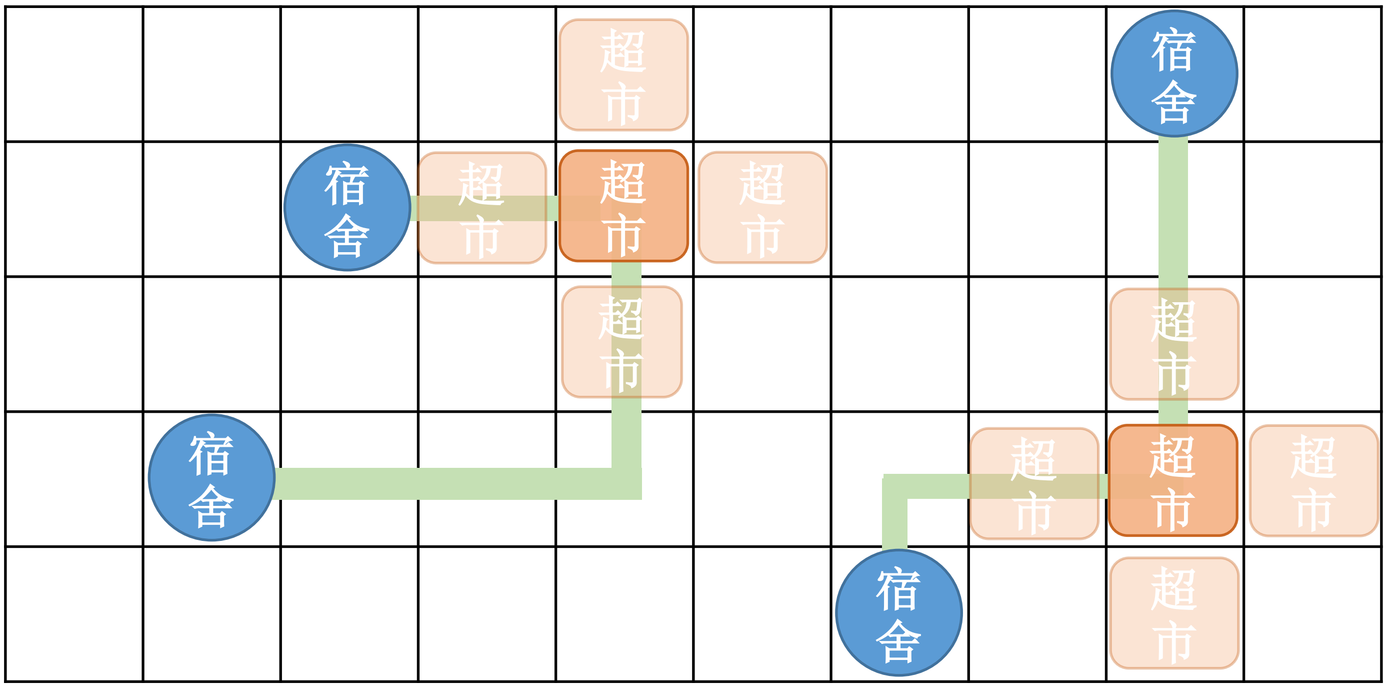
从7种该类更改方案中选择对应目标函数最低的方案



选址方案对应的目标函数值从15降低为13

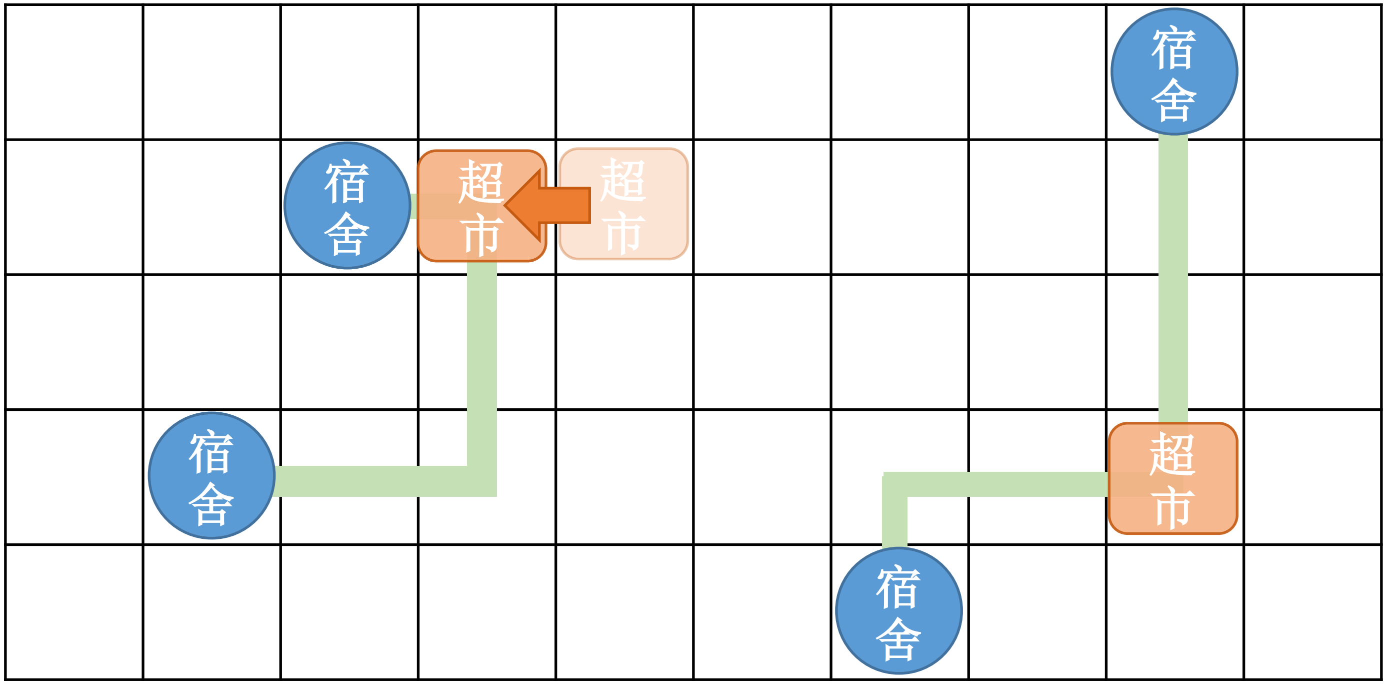
最优化方法示例

针对新方案又有8种更改方案
（仍仅考虑将一个超市移动一格的更改方案）



最优化方法示例

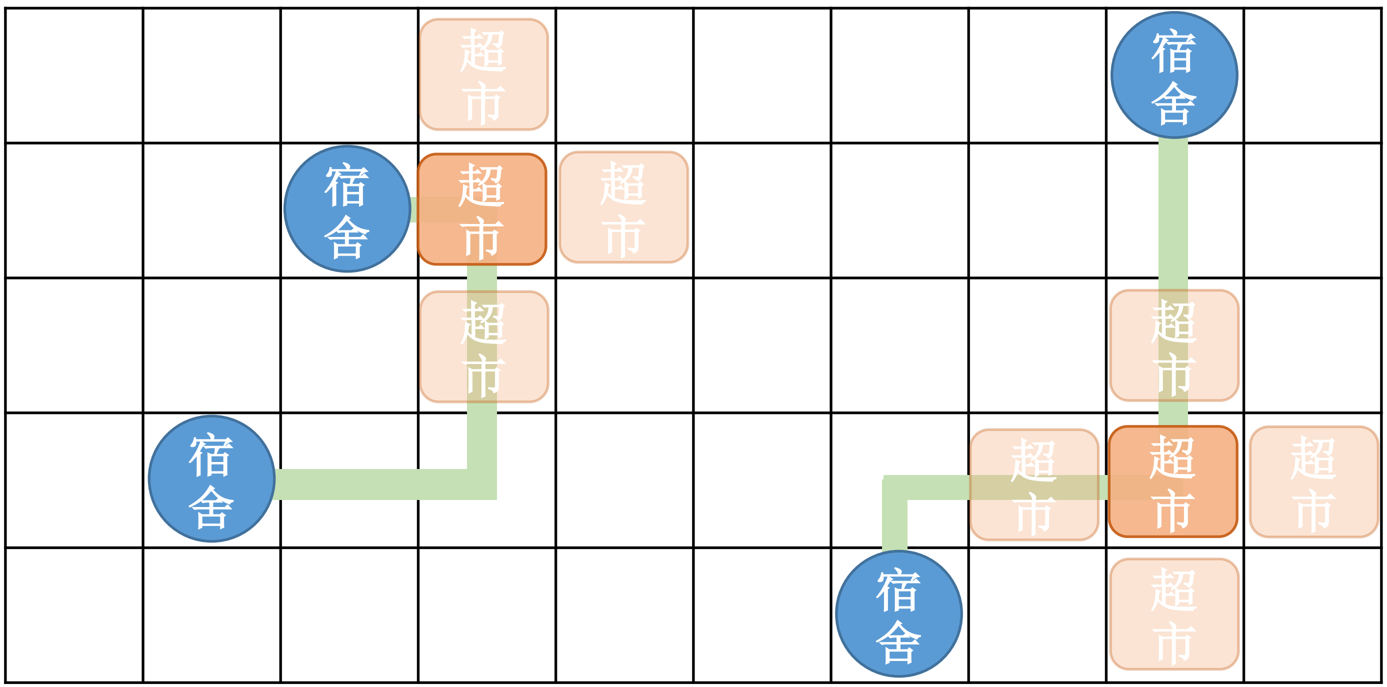
从8种该类更改方案中选择对应目标函数最低的方案



选址方案对应的目标函数值从13降低为11

最优化方法示例

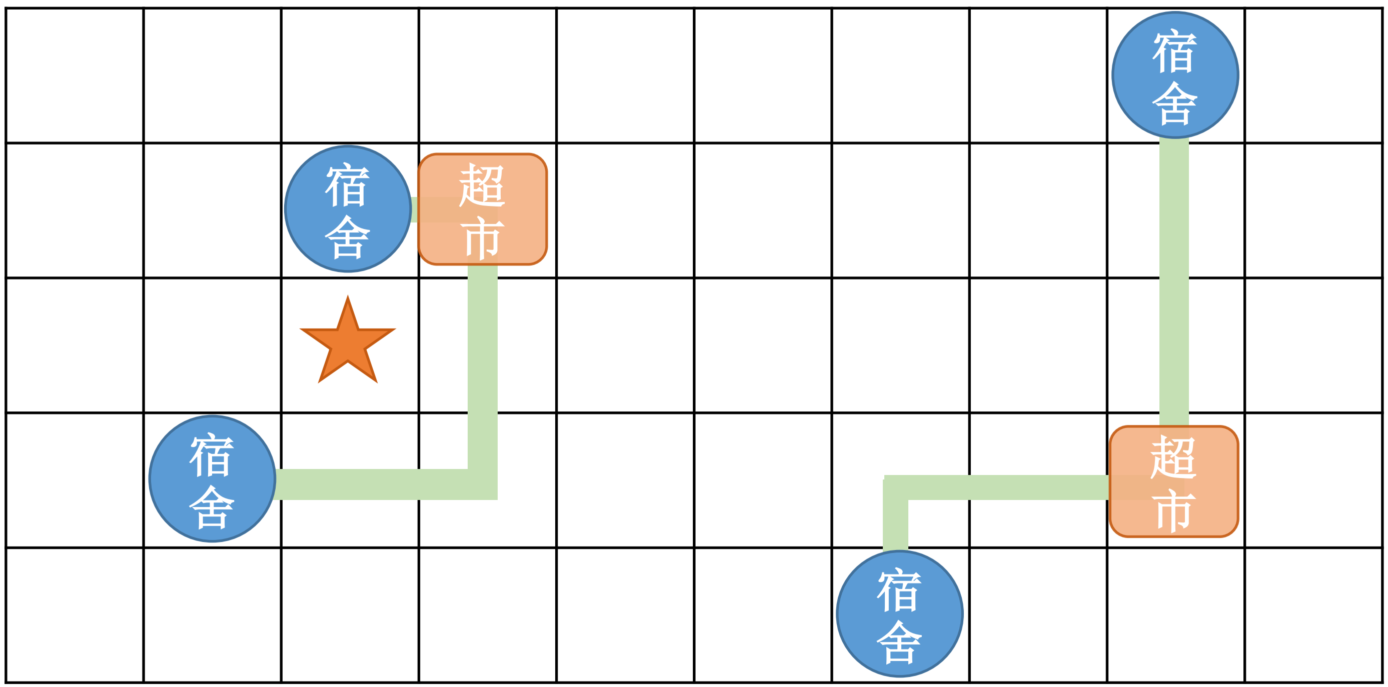
针对新方案有7种更改方案，但都不能进一步降低目标函数



该方法停止搜索，输出结果，对应目标函数值为11

最优化方法示例

该算法是否找到了最优结果?
该算法优劣是什么? 是否存在更好的算法?



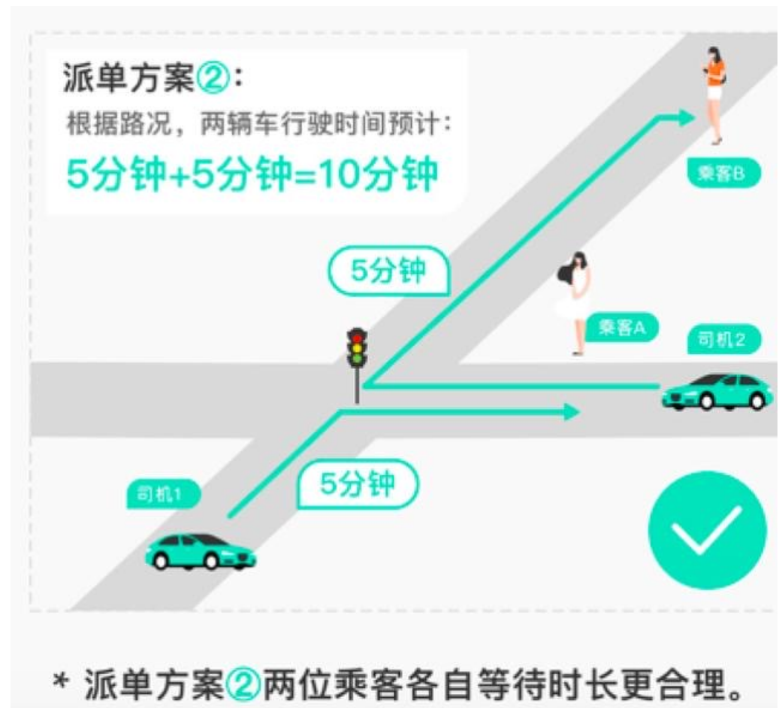
最优化问题

最优化问题广泛存在于各类应用场景

网约车APP

通过司乘匹配最小化总等待时长

（考虑优先级/…）



最优化问题

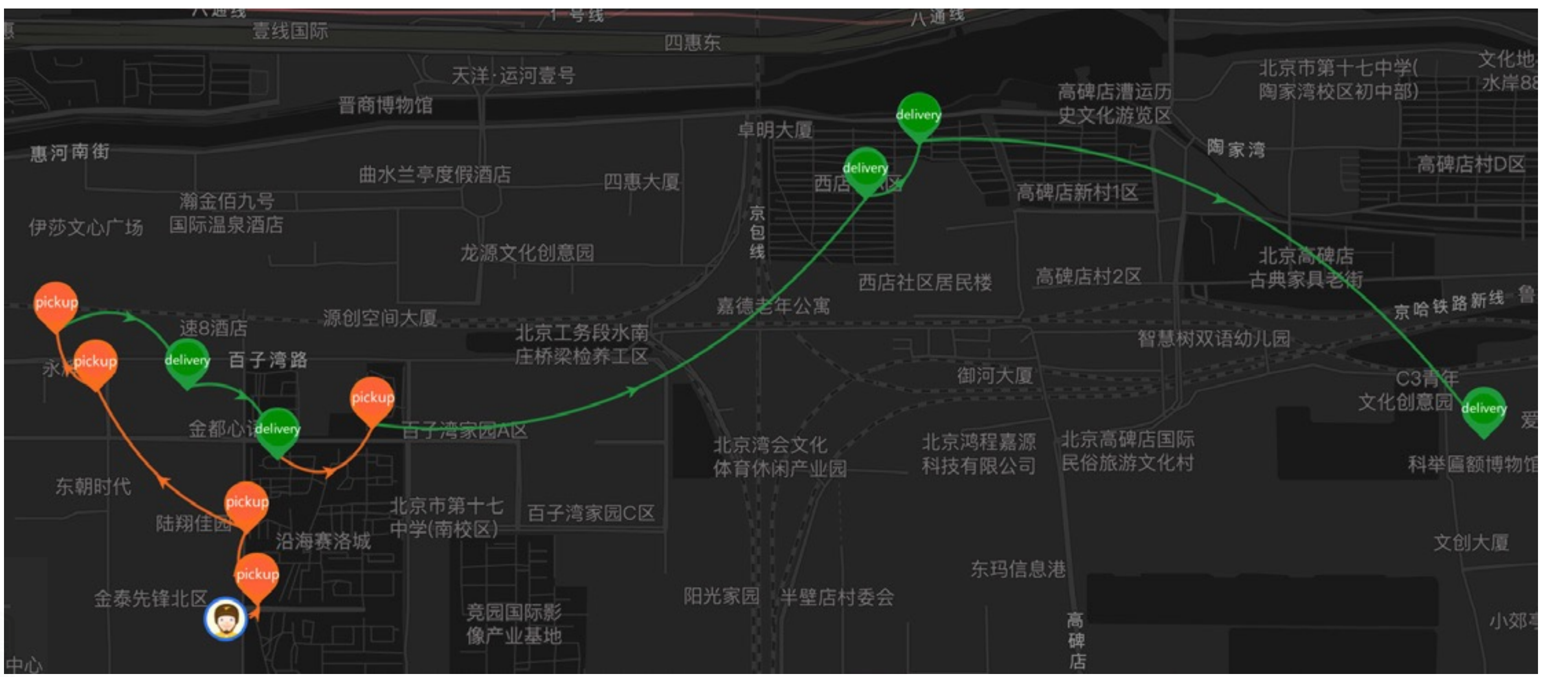
最优化问题广泛存在于各类应用场景

饿了么

外卖APP

通过选择取餐/送餐顺序最小化总路程

（考虑顾客等待时间/出餐时间/…）



最优化问题

最优化问题广泛存在于各类应用场景

通讯APP

通过学习神经网络参数最小化损失函数（翻译结果与真实标签的距离）

最优化问题

最优化问题广泛存在于各类应用场景

出行APP

通过**动态定价**最大化**总利润**

（考虑空房/空座量、未来需求量）

最优化问题

- 最优化问题广泛存在于各类应用场景
- 本课程：讲授解决各类最优化问题的各类方法

课程大纲

因与《数据结构与算法设计》内容有重叠，略讲

Part1 如何从复杂度的角度衡量算法好坏？
算法复杂度、**P**问题与**NP**问题与**NP**难问题
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
无约束凸优化问题、梯度下降法、牛顿法； 线性规划问题、单纯形法、内点法； 有约束凸优化问题、**KKT**条件、内点法
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
局部搜索、模拟退火、遗传算法、差分进化算法、蚁群算法、粒子群优化算法、人工蜂群算法
Part4 针对较复杂的多目标最优化问题，如何找到较好解？
多目标优化问题、**NSGA-II**算法

算法复杂度

Part1 如何从复杂度的角度衡量算法好坏？
算法复杂度、P问题与NP问题与NP难问题
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

算法复杂度

- 从计算的角度衡量一个计算机算法的效率

- 包括时间和空间复杂度：

  ① 时间复杂度 → 计算所需的步数或指令条数

  ② 空间复杂度 → 计算所需的存储空间大小

算法复杂度

- 算法时间复杂度是关于问题规模**n**（或算法输入的大小**n**）的函数，描述计算所需步数随**n**增长的速度

- 什么是问题规模/输入大小？

排序问题: 为n个数进行排序

旅行商问题: 确定访问n个地点的顺序

线性规划问题: 解决包含n个变量、m个约束条件的线性规划问题

– 将算法时间复杂度写成关于**n**和**m**的函数

– 或者考察当**n**（或**m**）固定时，关于**m**（或**n**）的变化情况

算法复杂度

算法复杂度用大O符号表示，描述当n趋近于无穷时，复杂度的增长情况

例如，当问题规模为n时，某算法最多需要$5n^3+3n$的计算步数运行完毕，那么该算法的时间复杂度是$O(n^3)$

算法复杂度

算法复杂度用大O符号表示，描述当n趋近于无穷时，复杂度的增长情况

例如，当问题规模为n时，某算法最多需要$5n^3+3n$的计算步数运行完毕，那么该算法的时间复杂度是$O(n^3)$

可以直接写作： $5n^3+3n=O(n^3)$, 当n趋近于∞

具体定义

if there exists a positive real number M and a real number x_0

$$|f(x)| \leq Mg(x) \quad \text{for all } x \geq x_0.$$

then we can write: $f(x) = O\big(g(x)\big) \quad \text{as } x \to \infty$

算法复杂度

算法复杂度用大O符号表示，描述当n趋近于无穷时，复杂度的增长情况

例如，当问题规模为n时，某算法最多需要$5n^3+3n$的计算步数运行完毕，那么该算法的时间复杂度是$O(n^3)$

可以直接写作： $5n^3+3n=O(n^3)$, 当n趋近于∞

具体定义

if there exists a positive real number M and a real number x_0

$$|f(x)| \leq Mg(x) \quad \text{for all } x \geq x_0.$$

then we can write: $f(x) = O\big(g(x)\big) \quad \text{as } x \to \infty$

例如，可以写 $5n^2+n\log n=O(n^2)$, 当n趋近于∞

算法复杂度

Comparison sorts [edit]

空间复杂度

Below is a table of comparison sorts. A comparison sort cannot perform better than $O(n \log n)$.[4]

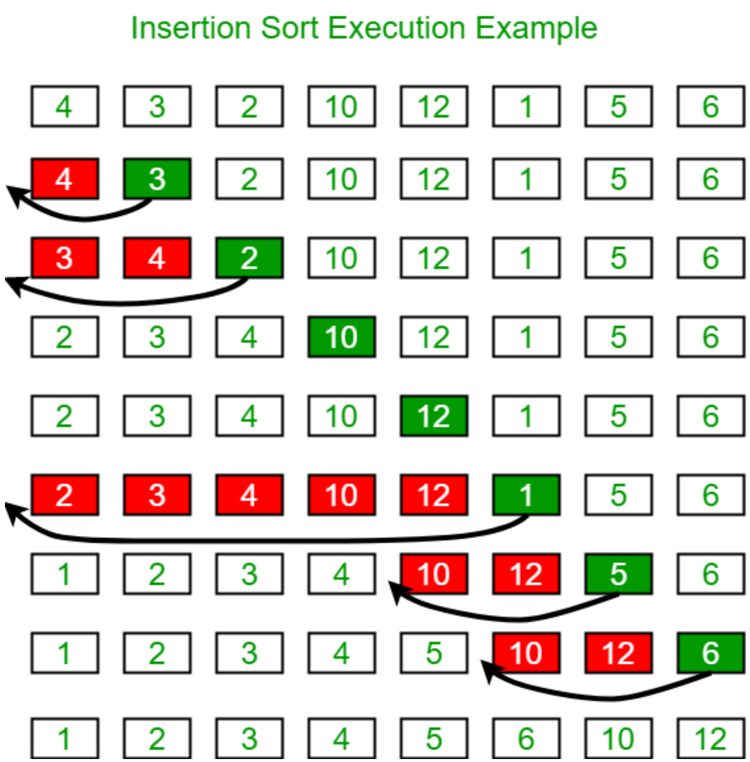
为什么有不同的时间复杂度？

Comparison sorts

Name	Best	Average	Worst	Memory	Stable	Method
Quicksort	$n \log n$	$n \log n$	n^2	$\log n$	No	Partitioning
Merge sort	$n \log n$	$n \log n$	$n \log n$	n	Yes	Merging
In-place merge sort	—	—	$n \log^2 n$	1	Yes	Merging
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	No	Partitioning & Selection
Heapsort	$n \log n$	$n \log n$	$n \log n$	1	No	Selection
Insertion sort	n	n^2	n^2	1	Yes	Insertion
Block sort	n	$n \log n$	$n \log n$	1	Yes	Insertion & Merging
Quadsort	n	$n \log n$	$n \log n$	n	Yes	Merging
Timsort	n	$n \log n$	$n \log n$	n	Yes	Insertion & Merging
Selection sort	n^2	n^2	n^2	1	No	Selection

算法复杂度

插入排序的时间算法复杂度？




```
for i = 1 to n-1
    key = A[i]
    j = i - 1
    while j >= 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    end while
A[j + 1] = key
end for
```

算法复杂度

插入排序的时间算法复杂度？

```
1. for i = 1 to n-1
2.     key = A[i]
3.     j = i - 1
4.     while j >= 0 and A[j] > key
5.         A[j + 1] = A[j]
6.         j = j - 1
7.     end while
8. A[j + 1] = key
9. end for
```

时间消耗	指令执行次数
C_1	n - 1
C_2	n - 1
C_3	n - 1
C_4	$\sum_{i=1}^{n-1}(1+t_i)$
C_5	$\sum_{i=1}^{n-1} t_i$
C_6	$\sum_{i=1}^{n-1} t_i$
C_8	n - 1

t_i 是在外层循环的第i次循环中，内层循环的次数

例子修改自iq.opengenus.org “Time Complexity of Insertion Sort”

算法复杂度

插入排序的时间算法复杂度？

```
for i = 1 to n-1
    key = A[i]
    j = i - 1
    while j >= 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    end while
A[j + 1] = key
end for
```

时间消耗	指令执行次数
C_1	n - 1
C_2	n - 1
C_3	n - 1
C_4	$\sum_{i=1}^{n-1}(1+t_i)$
C_5	$\sum_{i=1}^{n-1}t_i$
C_6	$\sum_{i=1}^{n-1}t_i$
C_8	n - 1

t_i 是在外层循环的第i次循环中，内层循环的次数

best case: $t_i=0$

即 $(C_1+C_2+C_3+C_4+C_8)(n-1)$

worst case: $t_i=i$

即 $(C_1+C_2+C_3+C_8)(n-1)+C_4(2+n)(n-1)/2+(C_5+C_6)(n-1)n/2$

算法复杂度

Name	Best	Average	Worst	Memory	Stable	Method
Insertion sort	n	n^2	n^2	1	Yes	Insertion

插入排序的时间算法复杂度？空间复杂度？

```
for i = 1 to n-1
    key = A[i]
    j = i - 1
    while j >= 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    end while
A[j + 1] = key
end for
```

时间消耗	指令执行次数
C_1	n - 1
C_2	n - 1
C_3	n - 1
C_4	$\sum_{i=1}^{n-1}(1+t_i)$
C_5	$\sum_{i=1}^{n-1}t_i$
C_6	$\sum_{i=1}^{n-1}t_i$
C_8	n - 1

t_i 是在外层循环的第i次循环中，内层循环的次数

best case: $t_i=0$

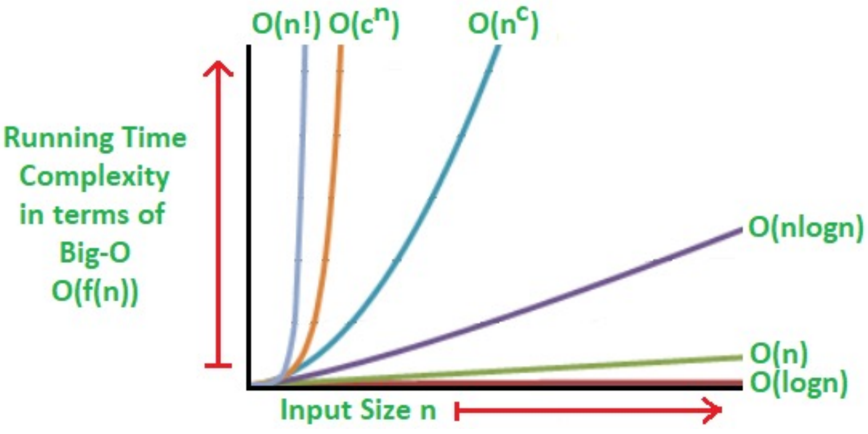
即 $(C_1+C_2+C_3+C_4+C_8)(n-1)$

worst case: $t_i=i$

即 $(C_1+C_2+C_3+C_8)(n-1)+C_4(2+n)(n-1)/2+(C_5+C_6)(n-1)n/2$

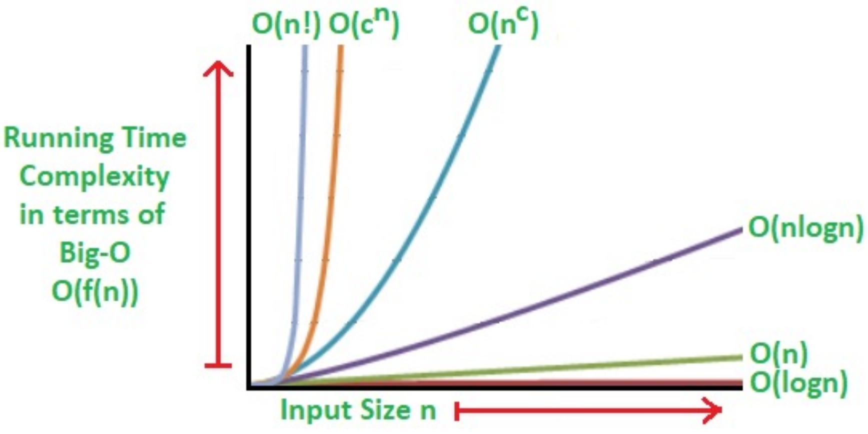
算法复杂度

O(*1*)	constant time
O(*log n*)	logarithmic time
O(*n*)	linear time
O(*n log n*)	linearithmic time
O(n^2)	quadratic time
O(n^c)	polynomial time
O(c^n)	exponential time
O(*n!*)	factorial time
O(∞)	infinite time



算法复杂度

O(*1*)	constant time
O(*log n*)	logarithmic time
O(*n*)	linear time
O(*n log n*)	linearithmic time
O(n^2)	quadratic time
O(n^c)	polynomial time
O(c^n)	exponential time
O(*n!*)	factorial time
O(∞)	infinite time



假设针对一个问题有四种算法，用每秒百万次的计算机运行

复杂性函数	问题规模n		
	10	30	60
n	0.01ms	0.03ms	0.06ms
n^3	1ms	27ms	216ms
n^5	100ms	24.3s	13min
2^n	1ms	17.9min	366世纪

P、NP、NP难问题

Part1 如何从复杂度的角度衡量算法好坏？
算法复杂度、P问题与NP问题与NP难问题
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

P问题

P: Polynomial time

可以找到一个在多项式时间之内（包含多项式时间）解决它的算法

$O(1)$	constant time
$O(\log n)$	logarithmic time
$O(n)$	linear time
$O(n \log n)$	linearithmic time
$O(n^2)$	quadratic time
$O(n^c)$	polynomial time
$O(c^n)$	exponential time
$O(n!)$	factorial time
$O(\infty)$	infinite time

P问题

例如：计算两个n位数的乘积

至少有一种算法的时间复杂度为$O(n^2)$
（其实还存在其他更快的算法）

```
       23958233
×          5830
———————————————
       00000000 ( =      23,958,233 ×     0)
      71874699  ( =      23,958,233 ×    30)
    191665864   ( =      23,958,233 ×   800)
+ 119791165     ( =      23,958,233 × 5,000)
———————————————
    139676498390 ( = 139,676,498,390        )
```

```
multiply(a[1..p], b[1..q], base)                    // Operands containing rightmost digits at index 1
  product = [1..p+q]                                // Allocate space for result
  for b_i = 1 to q                                  // for all digits in b
    carry = 0
    for a_i = 1 to p                                // for all digits in a
      product[a_i + b_i - 1] += carry + a[a_i] * b[b_i]
      carry = product[a_i + b_i - 1] / base
      product[a_i + b_i - 1] = product[a_i + b_i - 1] mod base
    product[b_i + p] = carry                        // last digit comes from final carry
  return product
```

P问题

例如：对n个数字排序

Comparison sorts [edit]

Below is a table of comparison sorts. A comparison sort cannot perform better than $O(n \log n)$.[4]

Comparison sorts

Name	Best	Average	Worst	Memory	Stable	Method
Quicksort	$n \log n$	$n \log n$	n^2	$\log n$	No	Partitioning
Merge sort	$n \log n$	$n \log n$	$n \log n$	n	Yes	Merging
In-place merge sort	—	—	$n \log^2 n$	1	Yes	Merging
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	No	Partitioning & Selection
Heapsort	$n \log n$	$n \log n$	$n \log n$	1	No	Selection
Insertion sort	n	n^2	n^2	1	Yes	Insertion
Block sort	n	$n \log n$	$n \log n$	1	Yes	Insertion & Merging
Quadsort	n	$n \log n$	$n \log n$	n	Yes	Merging
Timsort	n	$n \log n$	$n \log n$	n	Yes	Insertion & Merging
Selection sort	n^2	n^2	n^2	1	No	Selection

NP问题

- P问题：能在多项式的时间之内被解决

错误：NP问题指不能在多项式时间之内被解决的问题

- NP问题：能在多项式的时间之内验证一个解是否正确的问题

NP: Nondeterministic Polynomial

NP问题

- P问题：能在多项式的时间之内被解决

错误：NP问题指不能在多项式时间之内被解决的问题

- NP问题：能在多项式的时间之内验证一个解是否正确的问题

数独属于NP问题

NP问题

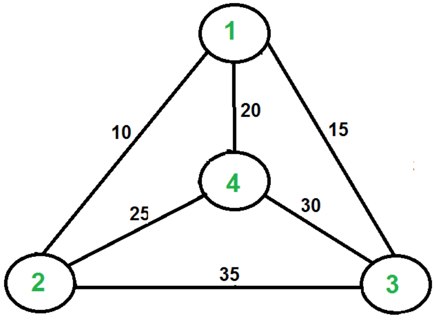
- P问题：能在多项式的时间之内被解决

错误：NP问题指不能在多项式时间之内被解决的问题

- NP问题：能在多项式的时间之内验证一个解是否正确的问题

旅行商问题的判定问题：给定n个城市，任意两个城市间有路相连且路程距离已知。找到一条总长度小于L的路径（从一个城市出发、不重复的遍历所有的城市并回到起点）

（旅行商问题的判定问题与旅行商问题不同）



NP问题

- P问题：能在多项式的时间之内被解决

错误：NP问题指不能在多项式时间之内被解决的问题

- NP问题：能在多项式的时间之内验证一个解是否正确的问题

两种问题之间的关系？

千禧年大奖难题

美国克雷研究所，每题100万美金

List of Unsolved Millennium Prize Problems

- P versus NP. P与NP问题
- Hodge conjecture. 霍奇猜想
- Riemann hypothesis. 黎曼猜想
- Yang–Mills existence and mass gap. 杨-米尔斯理论
- Navier–Stokes existence and smoothness. 纳维叶-斯托克斯方程
- Birch and Swinnerton-Dyer Conjecture. 贝赫和思维讷通-戴尔猜想
- ~~Poincaré conjecture.~~ 庞加莱猜想

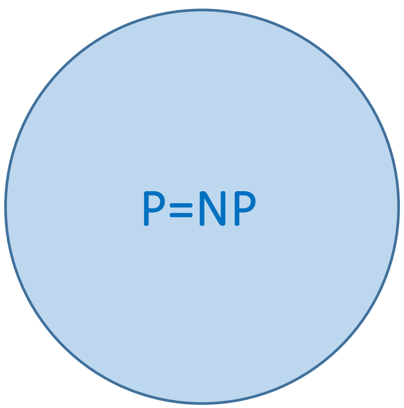
P vs NP

P问题：能在多项式时间内被解决

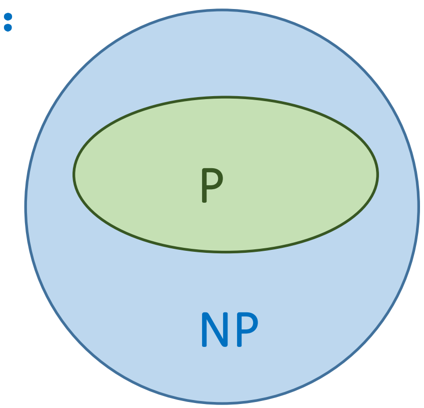
NP问题：能在多项式时间内验证解的有效性

易得所有的P问题都属于NP问题，是不是所有的NP问题也都属于P问题？

如果是：



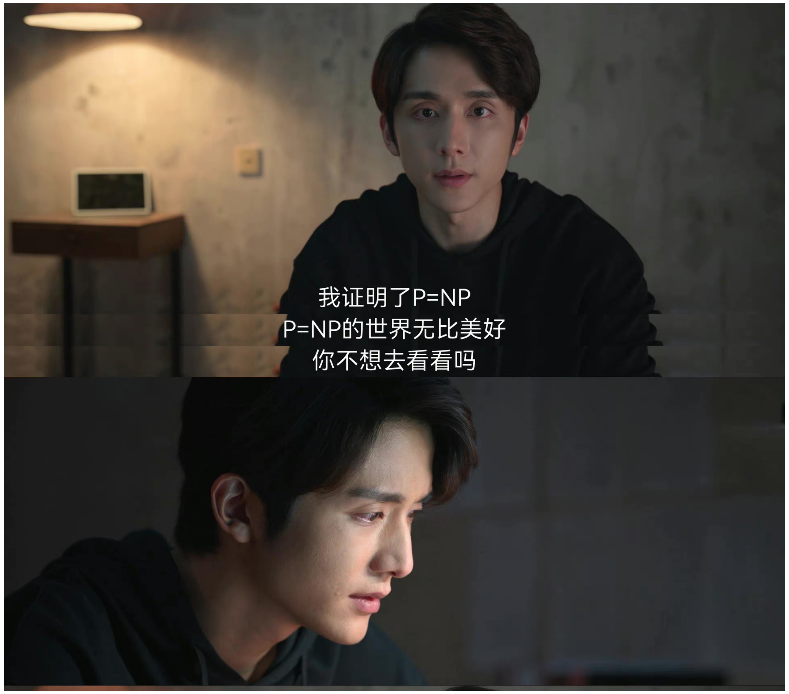
如果不是：



至今尚未被解决

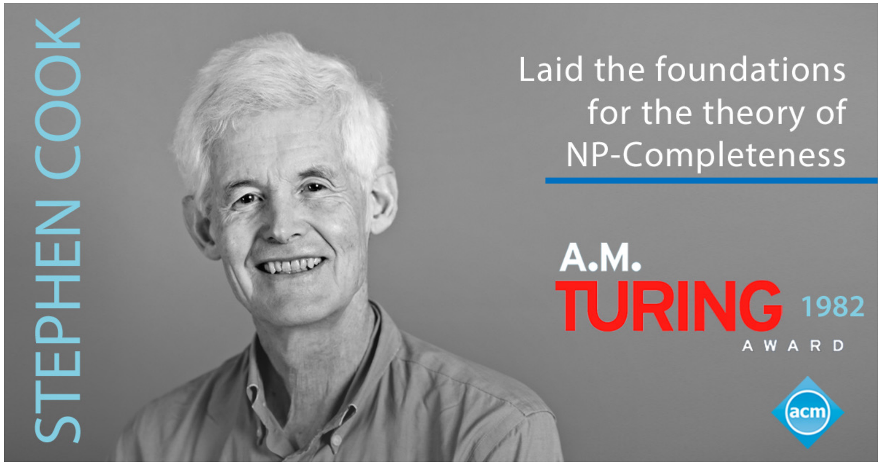
P vs NP


导演：沈严 编剧：聂成帅
THE HEART OF GENIUS
天才基本法
张子枫
饰
林朝夕
ZHANG
ZI
FENG
张新成
饰
裴之
ZHANG
XIN
CHENG
改编自晋江文学城小说《天才基本法》原著：长洱
今日会员收官
出品方
联合出品
@新浪电视

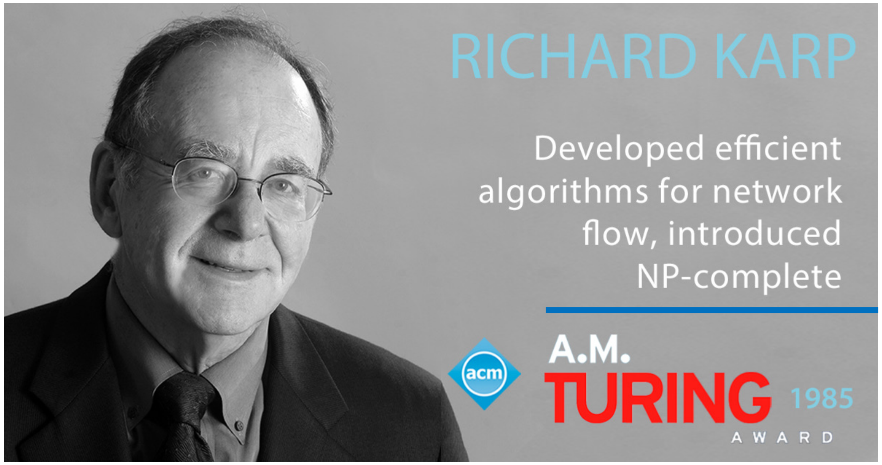

我证明了P=NP
P=NP的世界无比美好
你不想去看看吗

P vs NP

是不是所有的NP问题也都属于P问题？



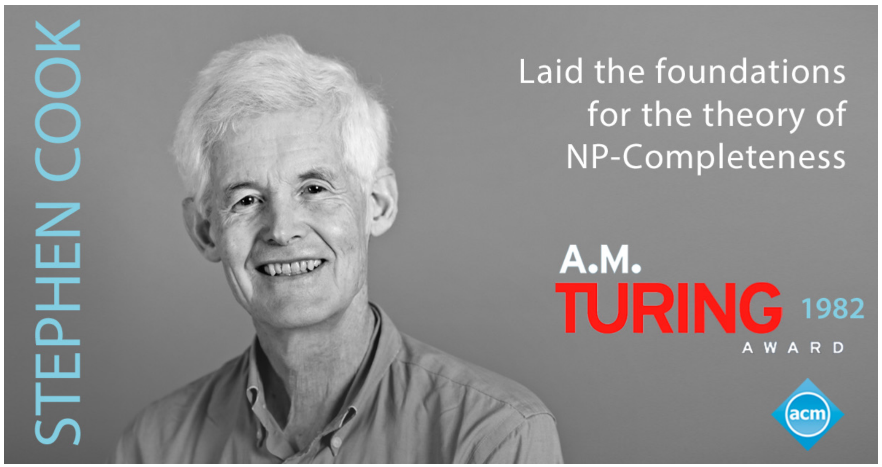
斯蒂芬库克

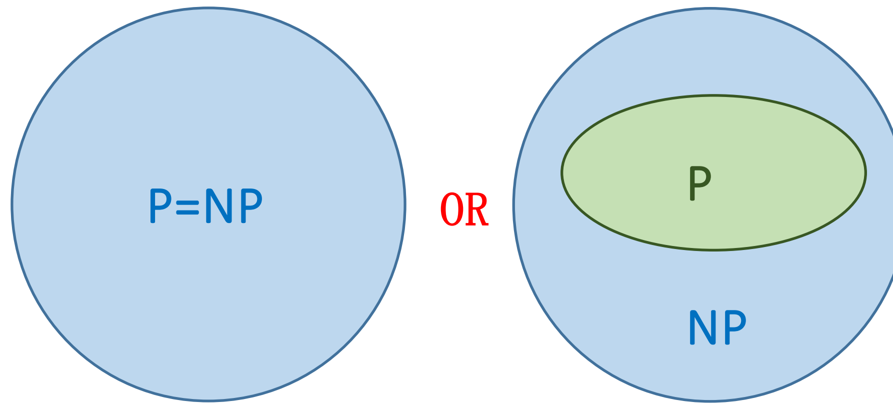


理查德卡普

NP完全问题

是不是所有的NP问题也都属于P问题？





斯蒂芬库克

思路：找到NP问题中最难的一类问题，试着证明它们是P问题，如果成立，那么可以说其余NP问题也是P问题

NP完全问题

思路：找到NP问题中最难的一类问题，试着证明它们是P问题，如果成立，那么可以说其余NP问题也是P问题

定义NP完全问题（NP-Complete Problems），即同时满足：

（1）NP问题；

（2）所有NP问题可在多项式时间内归约（be reducible to）到该问题

NP完全问题

归约：若A问题可以归约到B问题，那么B问题至少和A问题一样难，任何可以解决B问题的算法可以用于解决A问题

例子：对一元一次方程的求解问题可以归约到对二元一次方程组的求解问题

对于任意一元一次方程$a_1x+b_1=0$，能构造出二元一次方程组$m_{11}x+n_{11}y+k_{11}=0$和$m_{21}x+n_{21}y+k_{21}=0$，令$m_{11}-m_{21}=a_1$，$n_{11}-n_{21}=0$，k_{11}-$k_{21}=b_1$。

当求解到该二元一次方程组的解时，该解也是对应一元一次方程的解。

如：对$2x+3=0$可以构造$3x+y+6=0$与$x+y+3=0$

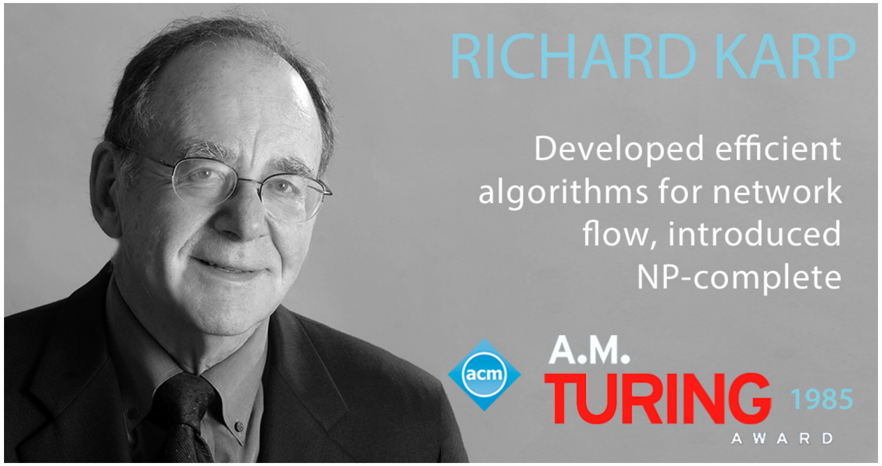
NP完全问题

P问题：能在多项式时间内被解决

NP问题：能在多项式时间内验证解的有效性

NP完全问题：所有NP问题可于多项式时间内归约到的NP问题

NP完全问题



理查德卡普

1972年，在“Reducibility Among Combinatorial Problems”中，提出21个NP完全问题

NP完全问题

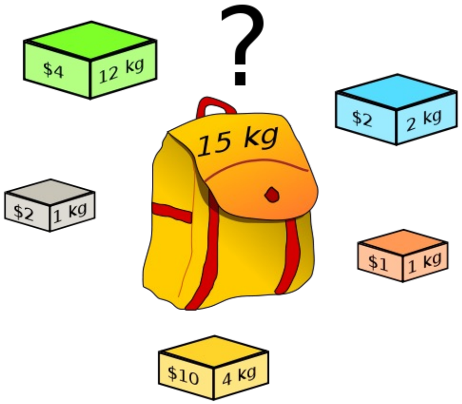
NP完全问题举例：

旅行商问题的判定问题：给定n个城市，任意两个城市间有路相连且路程距离已知。找到一条总长度小于L的路径（从一个城市出发、不重复的遍历所有的城市并回到起点）

NP完全问题

NP完全问题举例：

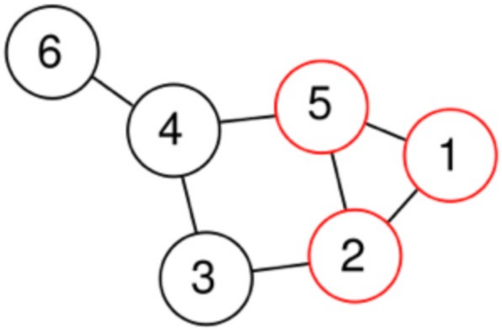
0-1背包问题的判定问题：给定一组物品，已知每个物品的重量和价值，问给定一个背包，是否有方案使得装进背包的物品的重量不超过W、价值超过V？



NP完全问题

NP完全问题举例：

分团问题的判定问题：给定一个图（包含若干顶点和边），是否有团使得团的顶点数目不小于K？



团：两两之间有边相连的顶点集合

NP完全问题

迄今，还没有证明任何NP完全问题属于P问题（即能在多项式时间内被解决）

如果能证明某NP完全问题属于P问题:

任何能在多项式时间内验证解的有效性的问题都能在多项式时间内被解决

整数规划、组合优化等难题有望被高效解决。现实中，大到公路、铁路、城市规划，小到提升送餐APP配送满意度等问题都有望被妥善的解决。

NP难问题

P问题：能在多项式时间内被解决

NP问题：能在多项式时间内验证解的有效性

NP完全（NP-Complete）问题：所有NP问题可于多项式时间内归约到的NP问题

NP难（NP-Hard）问题：所有NP问题可于多项式时间内归约到的问题

NP难问题 与 NP完全问题 对比

- NP完全问题是NP问题；NP难问题不用是NP问题
- NP完全问题一般是判定类问题，NP难问题不用是判定类问题

NP难问题

NP完全问题

旅行商问题的判定问题：给定n个城市，任意两个城市间有路相连且路程距离已知。找到一条总长度小于L的路径（从一个城市出发、不重复的遍历所有的城市并回到起点）

NP难问题

旅行商问题：给定n个城市，任意两个城市间有路相连且路程距离已知。找到总长度最小的路径（从一个城市出发、不重复的遍历所有的城市并回到起点）

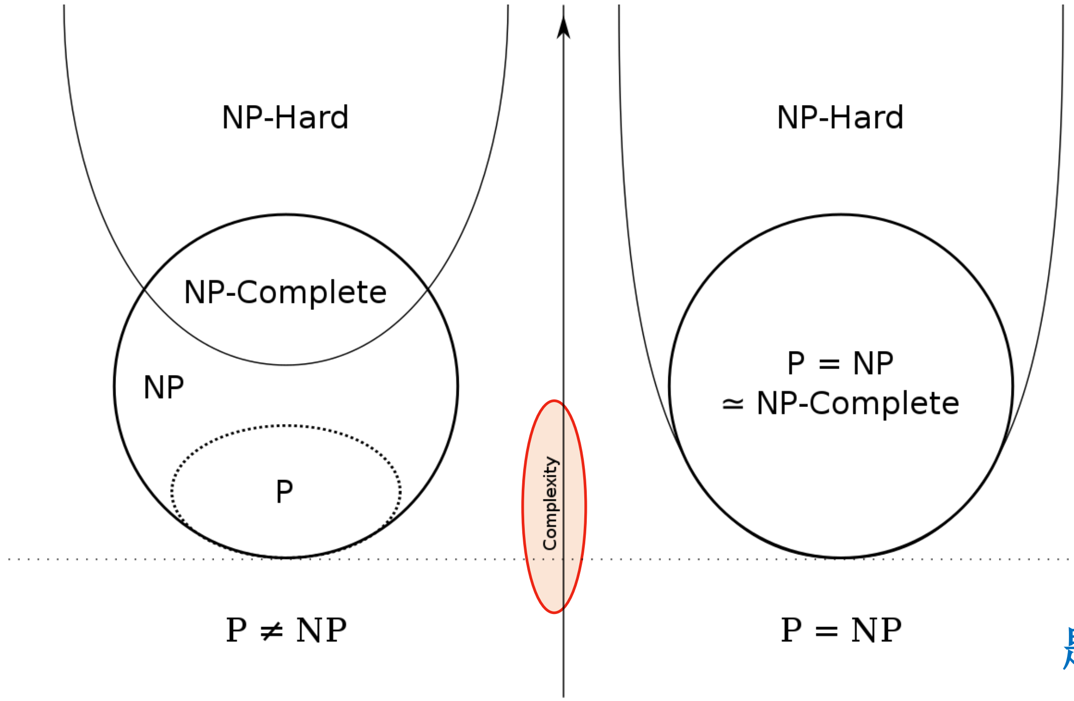
NP难问题

P问题：能在多项式时间内被解决

NP问题：能在多项式时间内验证解的有效性

NP完全问题：所有NP问题可于多项式时间内归约到的NP问题

NP难问题：所有NP问题可于多项式时间内归约到的问题



（不是所有的P问题都是NPC问题）

NP难问题

- 关于更多问题类别的定义

"世界七大数学难题之一: P与NP复杂度问题"
https://www.bilibili.com/video/av19085452?from=search&seid=13679117091167028644&spm_id_from=333.337.0.0

(7分55秒开始)

NP难问题

证明一个问题是NP-Complete或NP-Hard的目的是什么？

—— 公认接受通过智能优化算法找局部最优解来解决问题

最优化问题的一般数学形式

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
无约束凸优化问题、梯度下降法、牛顿法； 线性规划问题、单纯形法、内点法； 有约束凸优化问题、KKT条件、内点法
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

最优化问题

如何用数学符号表示最优化问题的一般形式

例如，一元二次方程的一般形式:

THE QUADRATIC FORMULA:

For any quadratic equation in the form $\boxed{ax^2 + bx + c = 0,}$ *the solution is*

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where : $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *are the coefficients of the* x^2 *and* x *terms, respectively*

$\boldsymbol{c}$ *is the constant*

例子

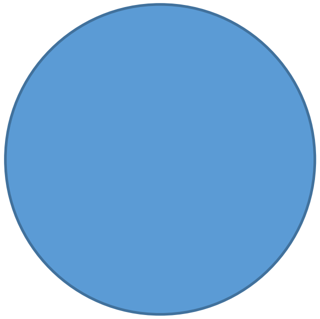
把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

例子

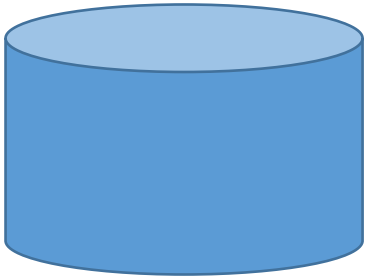
把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

用数学语言描述问题：定义参数和决策变量

实心金属球半径R



圆柱底半径 r、圆柱高h



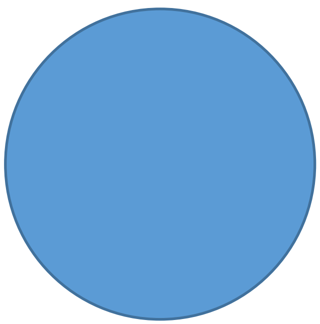
参数是固定的、用于描述问题：**R**

决策变量是可以调节的、用于描述对问题的解：**r**、**h**

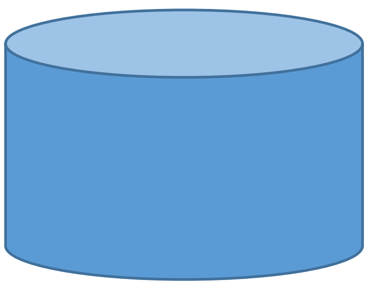
例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

实心金属球半径R



圆柱底半径 r、圆柱高h



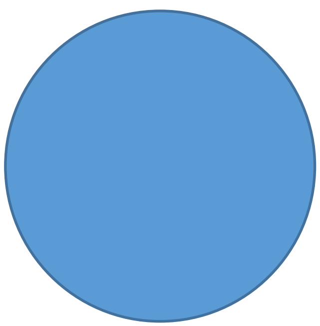
关键条件：金属球和圆柱体的质量相同 ⇒ 二者体积相同

$$\frac{4}{3}\pi R^3 = \pi r^2 h$$

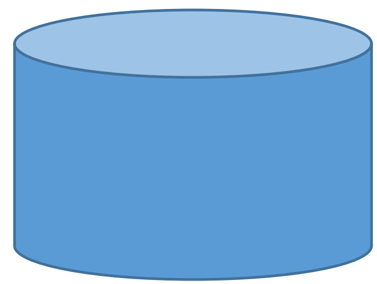
例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

实心金属球半径R



圆柱底半径 r、圆柱高h



目标：最小化圆柱体的表面积 $2\pi r^2 + 2\pi rh$

例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

对圆柱取尺寸的问题可以由如下数学问题表示：

minimize	$\min \quad 2\pi r^2 + 2\pi r h$	目标函数
subject to	$\text{s.t.} \quad \frac{4}{3}\pi R^3 = \pi r^2 h$	约束条件
variable(s)	$\text{var.} \quad r > 0, h > 0.$	决策变量

例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

$$
\begin{aligned}
\min \quad & 2\pi r^2 + 2\pi r h \\
\text{s.t.} \quad & \frac{4}{3}\pi R^3 = \pi r^2 h \\
\text{var.} \quad & r > 0, h > 0.
\end{aligned}
$$

把约束条件化简，可以得到r和h的关系

利用该关系将问题转化成一个关于单个决策变量的问题

通过求解得到答案

例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

$$\begin{aligned} \min \quad & 2\pi r^2 + 2\pi r h \\ \text{s.t.} \quad & \frac{4}{3}\pi R^3 = \pi r^2 h \\ \text{var.} \quad & r > 0, h > 0. \end{aligned}$$

$$\Rightarrow$$

$$\begin{aligned} \min \quad & 2\pi r^2 + 2\pi \frac{4R^3}{3r} \\ \text{var.} \quad & r > 0. \end{aligned}$$

分析新问题中目标函数关于r单调性（一阶导、二阶导）

可得目标函数关于r先减后增，易得最优r值

代回原约束得最优h值

例子

把一个实心金属球熔化后，铸成一个实心圆柱体，问圆柱体取什么尺寸才能使它的表面积最小?

$$\begin{aligned} \min \quad & 2\pi r^2 + 2\pi r h \\ \text{s.t.} \quad & \frac{4}{3}\pi R^3 = \pi r^2 h \\ \text{var.} \quad & r > 0, h > 0. \end{aligned}$$

$$\begin{aligned} \min \quad & 2\pi r^2 + 2\pi \frac{4R^3}{3r} \\ \text{var.} \quad & r > 0. \end{aligned}$$

$$r^* = \sqrt[3]{\frac{2}{3}}R, h^* = 2\cdot \sqrt[3]{\frac{2}{3}}R$$

最优化问题一般形式

最优化问题三要素

$$\begin{aligned} &\min && 2\pi r^2 + 2\pi r h \\ &\text{s.t.} && \frac{4}{3}\pi R^3 = \pi r^2 h \\ &\text{var.} && r > 0, h > 0. \end{aligned}$$

目标函数

约束条件

决策变量

最优化问题一般形式

最优化问题三要素

$$
\begin{aligned}
\min \quad & 2\pi r^2 + 2\pi r h \\
\text{s.t.} \quad & \frac{4}{3}\pi R^3 = \pi r^2 h \\
\text{var.} \quad & r > 0, h > 0.
\end{aligned}
$$

目标函数

约束条件

决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m, \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}.
\end{aligned}
$$

不等式约束

等式约束

可行域

$\boldsymbol{x}$ 是向量

一般（单目标）最优化问题都可整理成以上形式

最优化问题一般形式

目标函数	$\min \quad f(\boldsymbol{x})$
约束条件	$\text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m,$ 不等式约束
	$h_i(\boldsymbol{x}) = 0, j = 1, \dots, p,$ 等式约束
决策变量	$\text{var.} \quad \boldsymbol{x} \in \mathcal{D}.$ 可行域

注意1： 可行域（feasible region）有时可以化入约束条件，反之亦然

$$\begin{aligned} \min \quad & x + y^2 + xy \\ \text{s.t.} \quad & x^3 + y \ge 0, \\ \text{var.} \quad & x \ge 0, y \le 1. \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \min \quad & x + y^2 + xy \\ \text{s.t.} \quad & x^3 + y \ge 0, \\ & x \ge 0, \\ & y \le 1, \\ \text{var.} \quad & x, y. \end{aligned}$$

最优化问题一般形式

目标函数

约束条件

决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m, \text{ 不等式约束} \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \text{ 等式约束} \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \text{可行域}
\end{aligned}
$$

注意1： 可行域（feasible region）有时可以化入约束条件，反之亦然

$$
\begin{aligned}
\min \quad & x + y^2 + xy \\
\text{s.t.} \quad & x^3 + y \ge 0, \\
\text{var.} \quad & x \ge 0, y \le 1.
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
\min \quad & x + y^2 + xy \\
\text{var.} \quad & (x, y) \in \mathcal{D}_0.
\end{aligned}
$$

$$\mathcal{D}_0 \triangleq \{(x, y): x \ge 0, y \le 1, x^3 + y \ge 0\}$$

最优化问题一般形式

目标函数

约束条件

决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m, \text{ 不等式约束} \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \text{ 等式约束} \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \text{可行域}
\end{aligned}
$$

注意2： 最大化问题和带有≥约束的问题都可以整理成一般形式

$$
\begin{aligned}
\max \quad & \log x + \frac{y}{z} \\
\text{s.t.} \quad & y^4 + z \ge x^2, \\
& xy = x + z \sin y, \\
\text{var.} \quad & x, y, z.
\end{aligned}
\quad \Longleftrightarrow
$$

最优化问题一般形式

目标函数

约束条件

决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \quad \text{不等式约束} \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \quad \text{等式约束} \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \text{可行域}
\end{aligned}
$$

注意2： 最大化问题和带有≥约束的问题都可以整理成一般形式

$$
\begin{aligned}
\max \quad & \log x + \frac{y}{z} \\
\text{s.t.} \quad & y^4 + z \geq x^2, \\
& xy = x + z \sin y, \\
\text{var.} \quad & x, y, z.
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
f(\boldsymbol{x}) &\triangleq -\log x_1 - \frac{x_2}{x_3} \\
g(\boldsymbol{x}) &\triangleq x_1^2 - x_2^4 - x_3 \\
h(\boldsymbol{x}) &\triangleq x_1 x_2 - x_1 - x_3 \sin x_2
\end{aligned}
$$

最优化问题一般形式

目标函数
约束条件
决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i = 1, \ldots, m, \\
& h_i(\boldsymbol{x}) = 0, j = 1, \ldots, p, \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}.
\end{aligned}
$$

不等式约束

等式约束

可行域

为什么要转化成一般形式？

方便判断最优化问题的类别（凸优化、线性规划、二次规划等）

方便选择求解方法、实施算法

最优化问题一般形式

目标函数	$\min \quad f(\boldsymbol{x})$
约束条件	$\text{s.t.} \quad g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m,$ 不等式约束
	$h_i(\boldsymbol{x}) = 0, j = 1, \dots, p,$ 等式约束
决策变量	$\text{var.} \quad \boldsymbol{x} \in \mathcal{D}.$ 可行域

常用概念

可行解（**feasible solution**）：满足约束条件和可行域的解

最优解（**optimal solution**）：令目标函数达到最小值的可行解

全局最优解（**global optimal solution**）、局部最优解（**local optimal solution**）

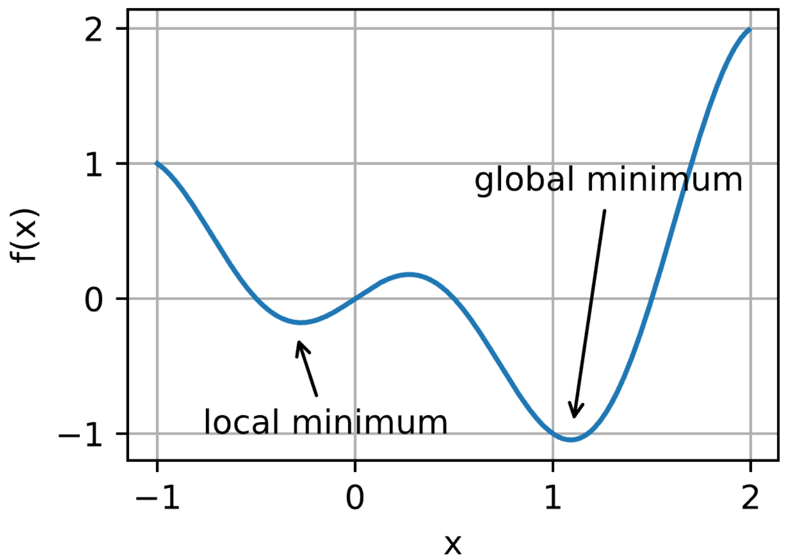
全局最优解$\boldsymbol{x}^*$：对所有可行解$\boldsymbol{x}$，有$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$

局部最优解$\boldsymbol{x}^*$：存在$\delta > 0$，使对所有满足$\|\boldsymbol{x} - \boldsymbol{x}^*\| < \delta$的可行解$\boldsymbol{x}$，有$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$

最优化问题一般形式

全局最优解$\boldsymbol{x}^*$：对所有可行解$\boldsymbol{x}$，有$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$

局部最优解$\boldsymbol{x}^*$：存在$\delta > 0$，使对所有满足$\|\boldsymbol{x} - \boldsymbol{x}^*\| < \delta$的可行解$\boldsymbol{x}$，有$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$



最优化问题一般形式

目标函数

约束条件

决策变量

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \quad \text{不等式约束} \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \quad \text{等式约束} \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \text{可行域}
\end{aligned}
$$

常用概念

可行解（**feasible solution**）：满足约束条件和可行域的解

最优解（**optimal solution**）：令目标函数达到最小值的可行解

全局最优解（**global optimal solution**）、局部最优解（**local optimal solution**）

一般情况下，“最优解”指的是“全局最优解”

注意：最优解一定是可行解，可行解不一定是最优解

无约束凸优化问题

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
无约束凸优化问题、梯度下降法、牛顿法； **线性规划问题、单纯形法、内点法；** **有约束凸优化问题、KKT条件、内点法**
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

无约束凸优化问题

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m, \\ & h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\ \text{var.} \quad & \boldsymbol{x} \in \mathcal{D}. \end{aligned}$$

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{var.} \quad & \boldsymbol{x} \in \mathcal{R}^n. \end{aligned}$$

其中， $f(\boldsymbol{x})$是二阶连续可微的凸函数

无约束凸优化问题

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, \\
& h_i(\boldsymbol{x}) = 0, j = 1, \dots, p, \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{D}.
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) \\
\text{var.} \quad & \boldsymbol{x} \in \mathcal{R}^n.
\end{aligned}
$$

其中，$f(\boldsymbol{x})$是二阶连续可微的凸函数

什么是凸函数？

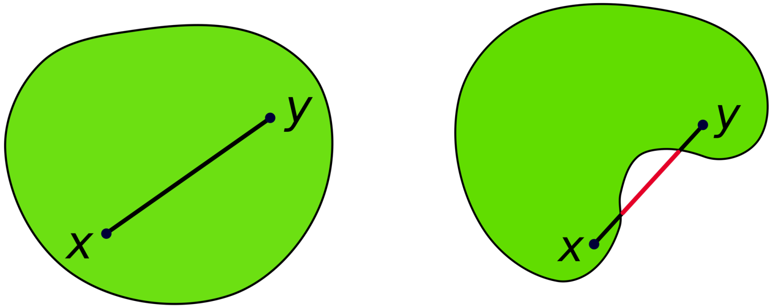
如何检验一个函数是否是凸函数？

例：$f(x) = x^2$

凸集与凸函数

凸集（convex set）

集合中任取两点，两点间线段上的点也属于集合

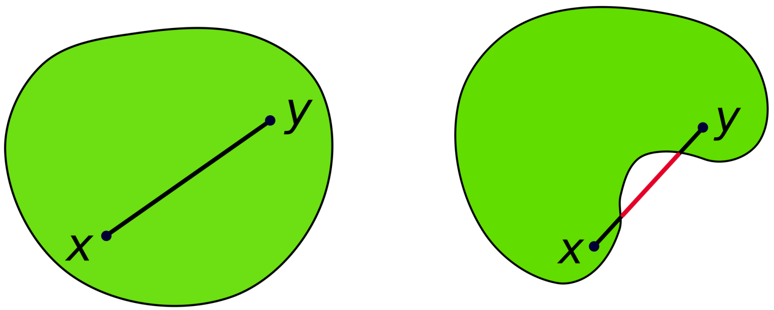


凸集　　　　非凸集

凸集与凸函数

凸集（convex set）

集合中任取两点，两点间线段上的点也属于集合



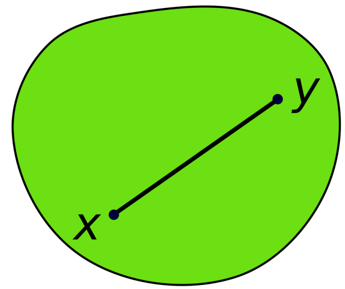
凸集　　非凸集

定义：若对任意$\boldsymbol{x},\boldsymbol{y} \in \mathcal{C}$及$\lambda \in [0,1]$，有$\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in \mathcal{C}$，则集合$\mathcal{C}$为凸集。

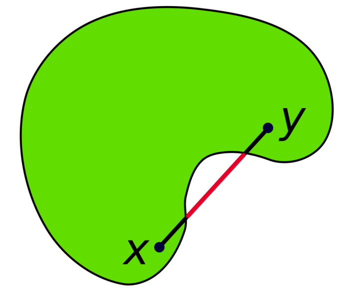
凸集与凸函数

凸集（convex set）

集合中任取两点，两点间线段上的点也属于集合



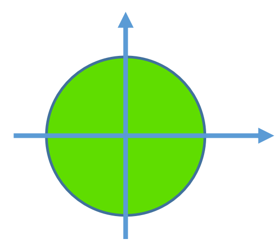
凸集



非凸集

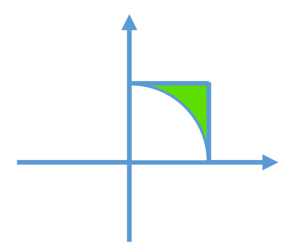
定义：若对任意$\boldsymbol{x}, \boldsymbol{y} \in C$及$\lambda \in [0,1]$，有$\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in C$，则集合$C$为凸集。

例：集合$\{(x_1, x_2) | x_1^2 + x_2^2 \leq 1\}$是凸集



（先利用完全平方公式证$x_1 y_1 + x_2 y_2 \leq 1$）

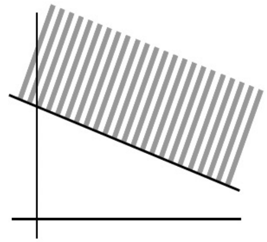
例：集合$\{(x_1, x_2) | x_1^2 + x_2^2 \geq 1, 0 \leq x_1, x_2 \leq 1\}$是非凸集



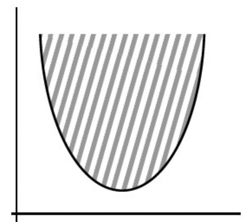
凸集与凸函数

凸函数

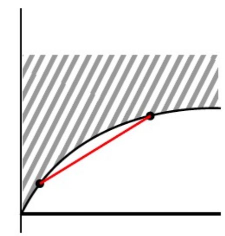
函数及其以上的区域是凸集



$f(x) = ax + b$



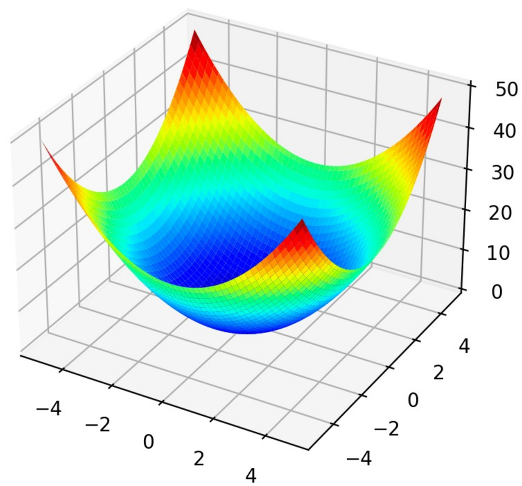
$f(x) = \text{quadratic}$

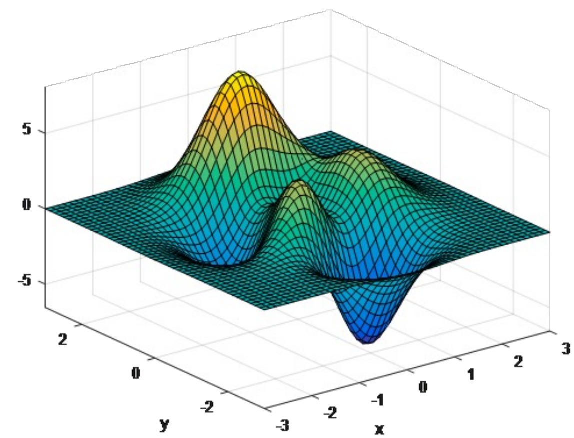


$f(x) = \sqrt{x}$

凸函数

非凸函数



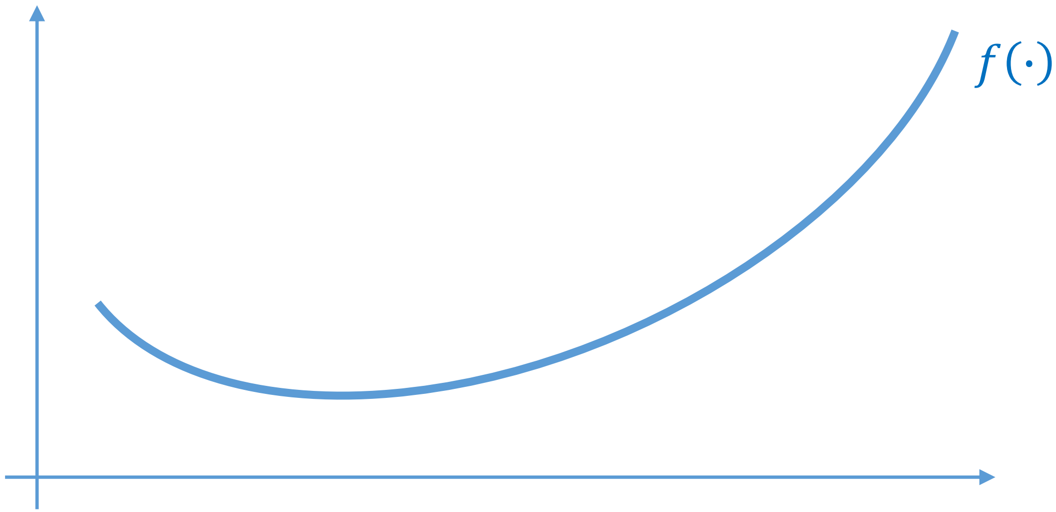


凸集与凸函数

凸函数定义：

（1）定义域$\mathcal{C}$是凸集；

（2）对任意$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$及$\lambda \in [0,1]$，$f(\cdot)$满足$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$。



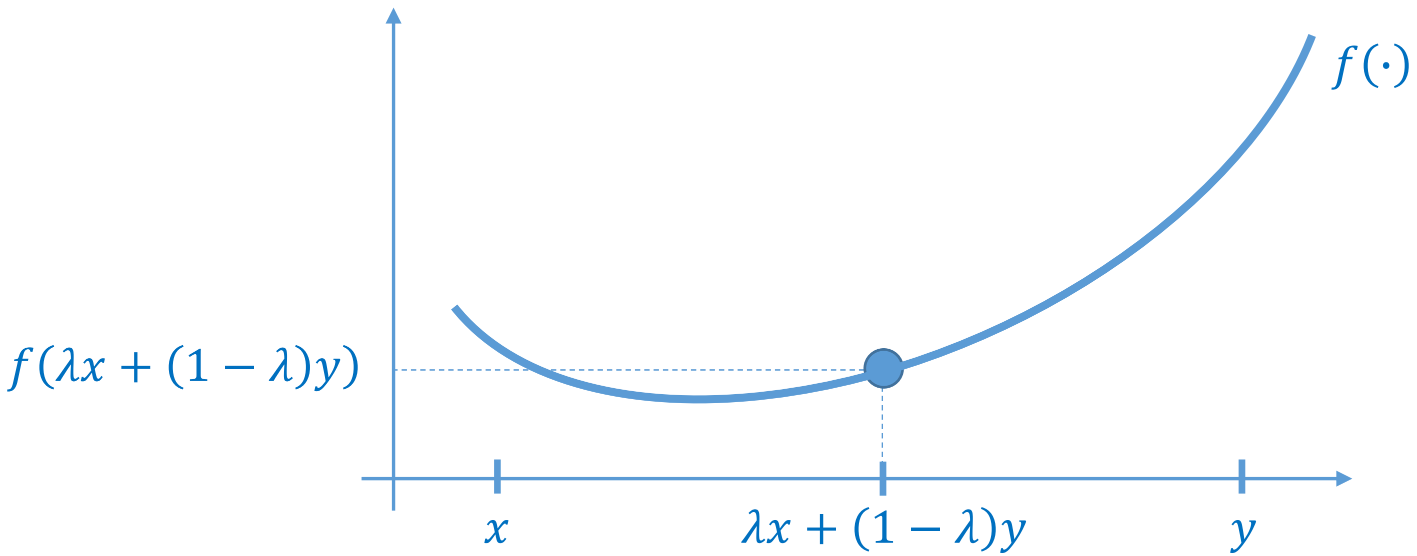
不等式在自变量是一维时的示意图

凸集与凸函数

凸函数定义：

（1）定义域C是凸集；

（2）对任意$\boldsymbol{x},\boldsymbol{y} \in C$及$\lambda \in [0,1]$，$f(\cdot)$满足$f(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \le \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$。



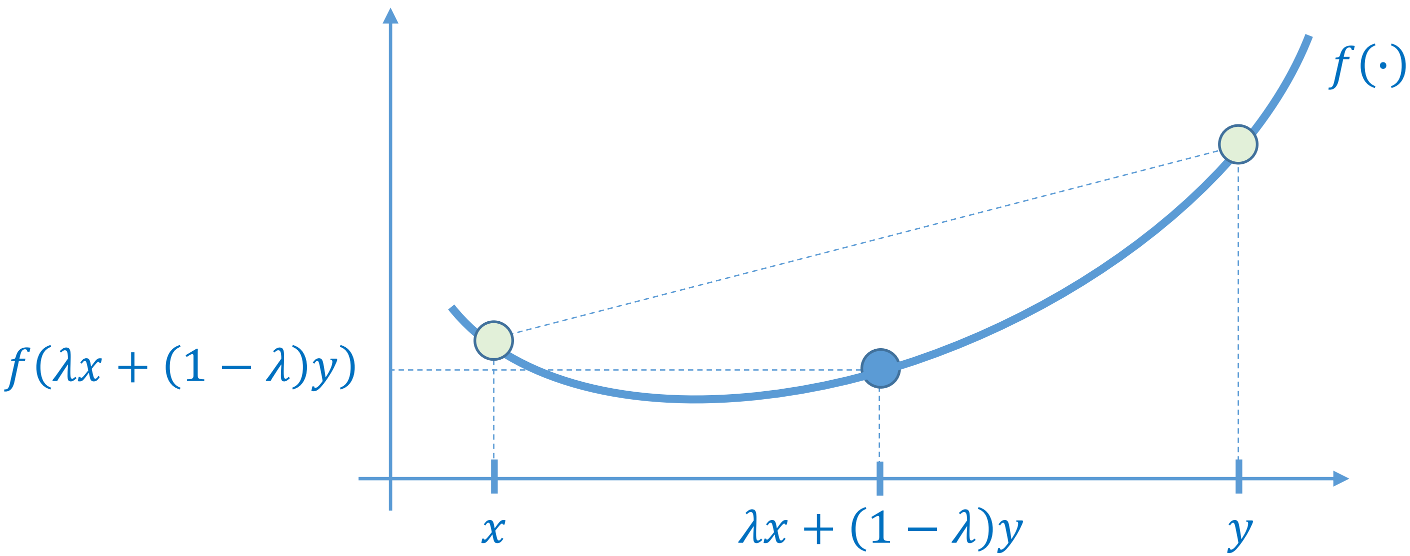
不等式在自变量是一维时的示意图

凸集与凸函数

凸函数定义：

（1）定义域C是凸集；

（2）对任意$\boldsymbol{x},\boldsymbol{y} \in C$及$\lambda \in [0,1]$，$f(\cdot)$满足$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \le \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$。



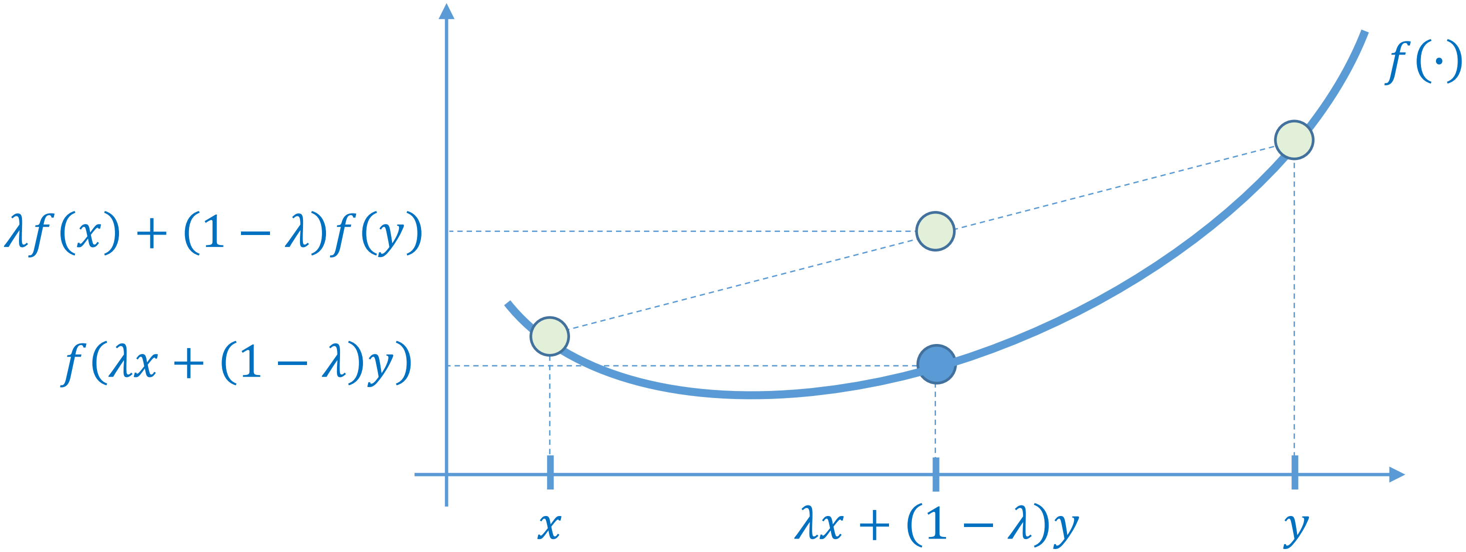
不等式在自变量是一维时的示意图

凸集与凸函数

凸函数定义：

（1）定义域C是凸集；

（2）对任意$\boldsymbol{x}, \boldsymbol{y} \in C$及$\lambda \in [0,1]$，$f(\cdot)$满足$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \le \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$。



不等式在自变量是一维时的示意图

凸集与凸函数

凸函数定义：

（1）定义域C是凸集；

（2）对任意$\boldsymbol{x},\boldsymbol{y}\in C$及$\lambda\in[0,1]$，$f(\cdot)$满足$f(\lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y})\le\lambda f(\boldsymbol{x})+(1-\lambda)f(\boldsymbol{y})$。

严格凸函数定义：

（1）定义域C是凸集；

（2）对任意$\boldsymbol{x},\boldsymbol{y}\in C$、$\boldsymbol{x}\neq\boldsymbol{y}$及$\lambda\in(0,1)$，有$f(\lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y})<\lambda f(\boldsymbol{x})+(1-\lambda)f(\boldsymbol{y})$。

凸函数例子：

$f(x)=x^2$

$f(x)=e^x$

$f(x)=-\log x, x>0$

$f(x)=x\log x, x>0$

如何快速检验一个函数是否是凸函数？

凸集与凸函数

定理（凸函数的一阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的一阶可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：

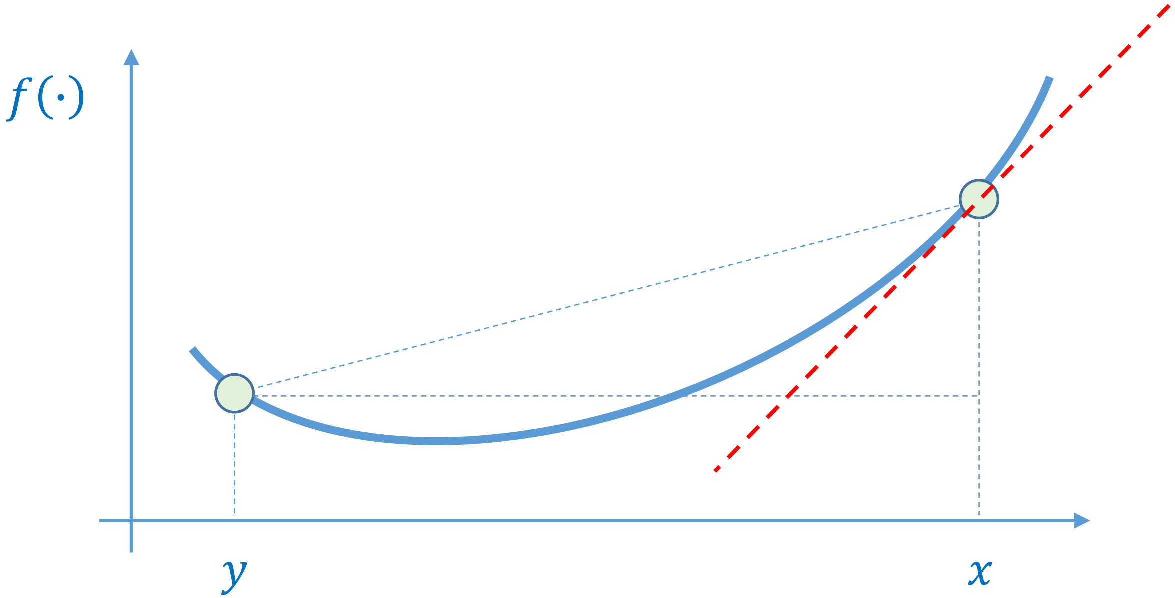
对任意$\boldsymbol{x}, \boldsymbol{y} \in C$，有$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x})$。

凸集与凸函数

定理（凸函数的一阶条件）

若$f(x)$是定义在凸集C上的一阶可微函数，那么$f(x)$是凸函数的**充要条件**是：

对任意$\boldsymbol{x},\boldsymbol{y}\in C$，有$f(\boldsymbol{y})\geq f(\boldsymbol{x})+\nabla f(\boldsymbol{x})^{T}(\boldsymbol{y}-\boldsymbol{x})$。



$x,y\in\mathcal{R}$且$x>y$时不等式可化简为$\frac{f(x)-f(y)}{x-y}\leq f'(x)$

凸集与凸函数

常用定理

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的充要条件是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

凸集与凸函数

常用定理

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的充要条件是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

海森矩阵 $$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

例，对$f(x_1, x_2)$

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f(x_1,x_2)}{\partial x_1^2} & \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x_1,x_2)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x_1,x_2)}{\partial x_2^2} \end{bmatrix}$$

凸集与凸函数

常用定理

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的充要条件是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \boldsymbol{0}$。

海森矩阵 $$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

例，对$f(x_1, x_2)$

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f(x_1,x_2)}{\partial x_1^2} & \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x_1,x_2)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x_1,x_2)}{\partial x_2^2} \end{bmatrix}$$

半正定矩阵（高等代数知识）

例：单位矩阵$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

对$n\times n$实对称矩阵$\mathbf{A}$，若对任意向量$\boldsymbol{u} \in \mathbb{R}^n$有$\boldsymbol{u}^T\mathbf{A}\boldsymbol{u} \geq 0$，$\mathbf{A}$为半正定矩阵。

凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

例，判断$f(x_1, x_2) = 4x_1^2 - 4x_1x_2 + 2x_2^2$是否是凸函数

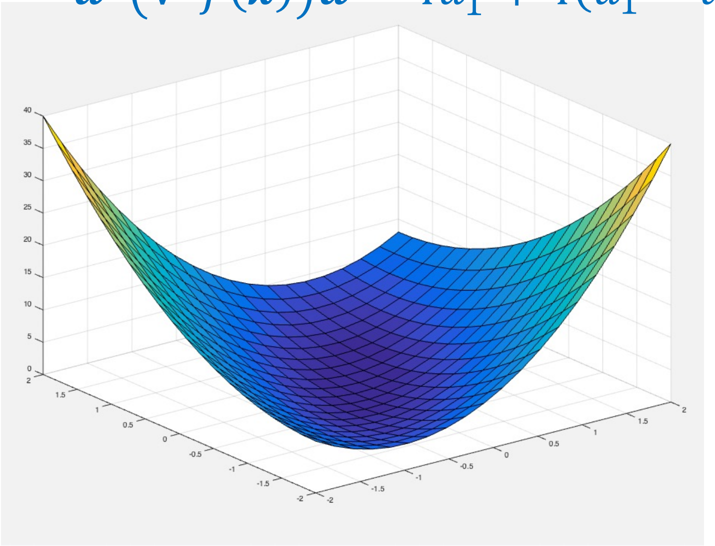
凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的充要条件是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \boldsymbol{0}$。

例，判断$f(x_1, x_2) = 4x_1^2 - 4x_1x_2 + 2x_2^2$是否是凸函数

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 8 & -4 \\ -4 & 4 \end{bmatrix} \qquad \boldsymbol{u}^T\left(\nabla^2 f(\boldsymbol{x})\right)\boldsymbol{u} = 4u_1^2 + 4(u_1 - u_2)^2 \geq 0$$



凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的充要条件是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

例，判断$f(x_1, x_2) = 4x_1^2 - 4x_1x_2 + 2x_2^2$是否是凸函数

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 8 & -4 \\ -4 & 4 \end{bmatrix} \qquad \boldsymbol{u}^T\left(\nabla^2 f(\boldsymbol{x})\right)\boldsymbol{u} = 4u_1^2 + 4(u_1 - u_2)^2 \geq 0$$

当变量数大于2时如何判断是否是半正定矩阵?

高等代数知识：对$n \times n$实对称矩阵$\mathbf{A}$，可以通过分析它的所有顺序主子式是否都非负，判断$\mathbf{A}$是否为半正定矩阵。

例，$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$的顺序主子式是$a_{11}$及$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$

凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \boldsymbol{0}$。

例，判断$f(x_1, x_2) = 4x_1^2 - 4x_1x_2 + 2x_2^2$是否是凸函数

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 8 & -4 \\ -4 & 4 \end{bmatrix} \qquad \boldsymbol{u}^T\left(\nabla^2 f(\boldsymbol{x})\right)\boldsymbol{u} = 4u_1^2 + 4(u_1 - u_2)^2 \geq 0$$

当变量数等于1时，条件 $\nabla^2 f(\boldsymbol{x}) \succcurlyeq \boldsymbol{0}$ 可化简为 $\frac{d^2 f(x)}{dx^2} \geq 0$

例：$f(x) = x^2$

凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

注意，该定理只适用于分析二阶连续可微函数

若$f(\boldsymbol{x})$不满足二阶连续可微，也可以是凸函数

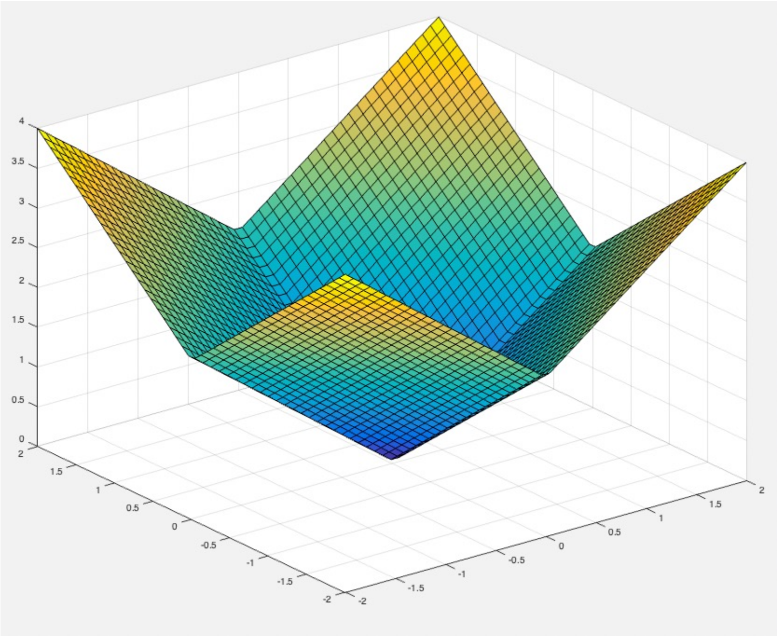
凸集与凸函数

定理（凸函数的二阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的二阶连续可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：对任意$\boldsymbol{x} \in C$，$f(\boldsymbol{x})$的海森矩阵是半正定矩阵，即$\nabla^2 f(\boldsymbol{x}) \succcurlyeq \mathbf{0}$。

注意，该定理只适用于分析二阶连续可微函数

若$f(\boldsymbol{x})$不满足二阶连续可微，也可以是凸函数



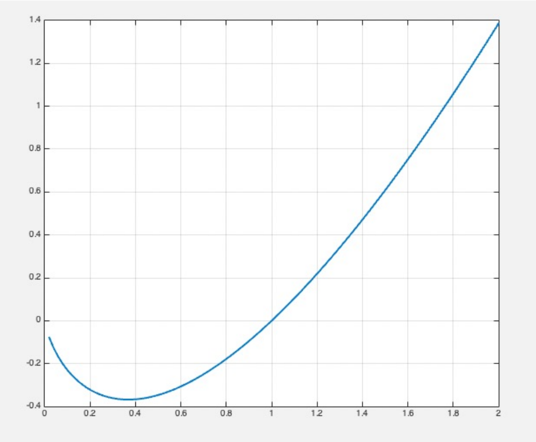
例：$f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1 = |x_1| + |x_2|$

无约束凸优化问题

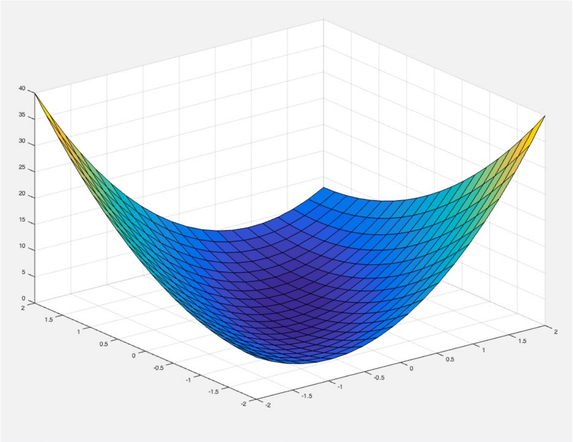
$$\min \quad f(\boldsymbol{x})$$
$$\text{var.} \quad \boldsymbol{x} \in \mathcal{R}^n.$$

其中，$f(\boldsymbol{x})$是二阶连续可微的凸函数

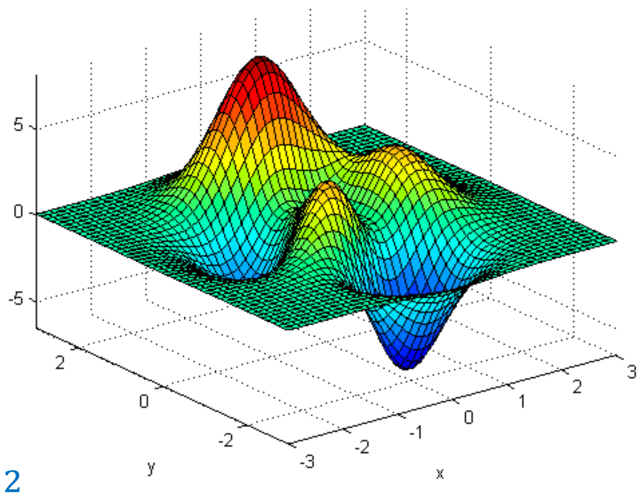
二阶连续可微凸函数
有什么性质可以简化问题求解?



$f(x) = x\log x, x > 0$



$f(x_1, x_2) = 4x_1^2 - 4x_1x_2 + 2x_2^2$



非凸函数

无约束凸优化问题

定理

不要求二阶连续可微

若$f(\boldsymbol{x})$是凸函数，那么任何使$f(\boldsymbol{x})$达到局部极小值的局部最优解$\boldsymbol{x}^{\text{local}}$也是使$f(\boldsymbol{x})$达到全局最小值的全局最优解。

Proof. We prove the result by contradiction. Let $\vec{x}_{\text{loc}}$ be a local minimum and $\vec{x}_{\text{glob}}$ a global minimum such that $f(\vec{x}_{\text{glob}}) < f(\vec{x}_{\text{loc}})$. Since $\vec{x}_{\text{loc}}$ is a local minimum, there exists $\gamma > 0$ for which $f(\vec{x}_{\text{loc}}) \leq f(\vec{x})$ for all $\vec{x} \in \mathbb{R}^n$ such that $\|\vec{x} - \vec{x}_{\text{loc}}\|_2 \leq \gamma$. If we choose $\theta \in (0,1)$ ~~small~~ large enough, $\vec{x}_\theta := \theta \vec{x}_{\text{loc}} + (1-\theta)\vec{x}_{\text{glob}}$ satisfies $\|\vec{x}_\theta - \vec{x}_{\text{loc}}\|_2 \leq \gamma$ and therefore

$$
\begin{aligned}
f(\vec{x}_{\text{loc}}) &\leq f(\vec{x}_\theta) && (5)\\
&\leq \theta f(\vec{x}_{\text{loc}}) + (1-\theta) f(\vec{x}_{\text{glob}}) \quad \text{by convexity of } f && (6)\\
&< f(\vec{x}_{\text{loc}}) \quad \text{because } f(\vec{x}_{\text{glob}}) < f(\vec{x}_{\text{loc}}). && (7)
\end{aligned}
$$

证明引自Carlos Fernandez-Granda <DS-GA1013/MATH-GA2821 Optimization-based Data Analysis>

无约束凸优化问题

定理

若$f(\boldsymbol{x})$是一阶可微凸函数，那么$\boldsymbol{x}^*$是最小化$f(\boldsymbol{x})$的全局最优解的充要条件是：

$$\nabla f(\boldsymbol{x}^*) = \mathbf{0}\ 。$$

无约束凸优化问题

定理

若$f(\boldsymbol{x})$是一阶可微凸函数，那么$\boldsymbol{x}^*$是最小化$f(\boldsymbol{x})$的全局最优解的充要条件是：

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0} \text{ 。}$$

当变量数等于1时，条件$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$可化简为 $\left.\frac{df(x)}{dx}\right|_{x=x^*} = 0$

无约束凸优化问题

定理

若$f(\boldsymbol{x})$是一阶可微凸函数，那么$\boldsymbol{x}^*$是最小化$f(\boldsymbol{x})$的全局最优解的**充要条件**是：

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}\ 。$$

可由凸函数的一阶条件证得：

定理（凸函数的一阶条件）

若$f(\boldsymbol{x})$是定义在凸集C上的一阶可微函数，那么$f(\boldsymbol{x})$是凸函数的**充要条件**是：

对任意$\boldsymbol{x}, \boldsymbol{y} \in C$，有$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x})$。

（充分性容易证，必要性的证明较难）

无约束凸优化问题

$$\min \quad f(\boldsymbol{x})$$
$$\text{var.} \quad \boldsymbol{x} \in \mathcal{R}^n.$$

其中，$f(\boldsymbol{x})$是二阶连续可微的凸函数

求解思路：搜索满足$\nabla f(\boldsymbol{x}^*) = \mathbf{0}$的$\boldsymbol{x}^*$

例：线性回归

（理工睿府）

面积	室	厅	建成时间	楼高（层）		售价
125.73	3	2	2014	20		540
85.25	2	1	2014	16		378
181.54	4	3	2014	20		620
…	…	…	…	…		…

假设 特征 与 售价 之间是线性函数关系，能否通过以上数据求出该线性函数？

例：线性回归

面积	室	厅	建成时间	楼高（层）	售价
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	b_1
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	b_2
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	b_3
…	…	…	…	…	…

M条

希望找到一组系数$\boldsymbol{w} = (w_1, w_2, w_3, w_4, w_5)^T$，使得:

$$
\begin{gathered}
w_1 a_{11} + w_2 a_{12} + w_3 a_{13} + w_4 a_{14} + w_5 a_{15} \to b_1 \\
w_1 a_{21} + w_2 a_{22} + w_3 a_{23} + w_4 a_{24} + w_5 a_{25} \to b_2 \\
w_1 a_{31} + w_2 a_{32} + w_3 a_{33} + w_4 a_{34} + w_5 a_{35} \to b_3 \\
\dots
\end{gathered}
$$

在实际的线性回归中，还包含常数项w_0
此外，特征$a_{11}, a_{21}, \dots$需要先做归一化处理

例：线性回归

希望找到一组系数$\boldsymbol{w}=(w_1,w_2,w_3,w_4,w_5)^T$，使得：

$$
\begin{aligned}
w_1a_{11}+w_2a_{12}+w_3a_{13}+w_4a_{14}+w_5a_{15} &\to b_1\\
w_1a_{21}+w_2a_{22}+w_3a_{23}+w_4a_{24}+w_5a_{25} &\to b_2\\
w_1a_{31}+w_2a_{32}+w_3a_{33}+w_4a_{34}+w_5a_{35} &\to b_3\\
&\cdots
\end{aligned}
$$

$\boldsymbol{Aw}\to\boldsymbol{b}$

线性回归（L2正则化）

$$
\begin{aligned}
\min \quad & \frac{1}{M}\|\boldsymbol{Aw}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2\\
\text{var.} \quad & \boldsymbol{w}\in\mathcal{R}^N.
\end{aligned}
$$

线性回归（L1正则化）

$$
\begin{aligned}
\min \quad & \frac{1}{M}\|\boldsymbol{Aw}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|\\
\text{var.} \quad & \boldsymbol{w}\in\mathcal{R}^N.
\end{aligned}
$$

LASSO

为避免过拟合，在目标函数中加入正则项：

$$\|\boldsymbol{w}\|_2^2=w_1^2+w_2^2+\cdots+w_N^2$$

$$\|\boldsymbol{w}\|=|w_1|+|w_2|+\cdots+|w_N|$$

（希望出现更多取值为零的系数）

例：线性回归

线性回归（L2正则化）

$$\min \quad \frac{1}{M}\|\boldsymbol{Aw}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2$$
$$\text{var.} \quad \boldsymbol{w}\in\mathcal{R}^N.$$

$$\|\boldsymbol{w}\|_2^2=w_1^2+w_2^2+\cdots+w_N^2$$

目标函数是否是二阶连续可微凸函数?

线性回归（L1正则化）

$$\min \quad \frac{1}{M}\|\boldsymbol{Aw}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|$$
$$\text{var.} \quad \boldsymbol{w}\in\mathcal{R}^N.$$

LASSO

$$\|\boldsymbol{w}\|=|w_1|+|w_2|+\cdots+|w_N|$$

凸函数：对任意$\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}$及$\lambda\in[0,1]$，满足$f(\lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y})\le\lambda f(\boldsymbol{x})+(1-\lambda)f(\boldsymbol{y})$

例：线性回归

线性回归（L2正则化）

$$\begin{aligned}\min\quad & \frac{1}{M}\|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2\\ \text{var.}\quad & \boldsymbol{w}\in\mathcal{R}^N.\end{aligned}$$

$$\|\boldsymbol{w}\|_2^2=w_1^2+w_2^2+\cdots+w_N^2$$

目标函数是否是二阶连续可微凸函数?

是二阶连续可微凸函数

线性回归（L1正则化）

$$\begin{aligned}\min\quad & \frac{1}{M}\|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|\\ \text{var.}\quad & \boldsymbol{w}\in\mathcal{R}^N.\end{aligned}$$

LASSO

$$\|\boldsymbol{w}\|=|w_1|+|w_2|+\cdots+|w_N|$$

是凸函数

证明要点:

1. 若$f_1(\boldsymbol{w})$和$f_2(\boldsymbol{w})$都为凸函数，那么$f_1(\boldsymbol{w})+f_2(\boldsymbol{w})$也是凸函数（利用定义）
2. $\|\boldsymbol{w}\|_2^2$是凸函数（分析海森矩阵）
3. $\|\boldsymbol{w}\|$是凸函数（利用定义）
4. $\|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2$是凸函数

半正定矩阵：$n \times n$实对称矩阵A，对任意向量$\boldsymbol{u} \in \mathcal{R}^n$有$\boldsymbol{u}^T \mathbf{A} \boldsymbol{u} \geq 0$

例：线性回归

线性回归（L2正则化）		
min	$\frac{1}{M}\|\mathbf{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2$
var.	$\boldsymbol{w} \in \mathcal{R}^N$.

线性回归（L1正则化）		
min	$\frac{1}{M}\|\mathbf{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|$
var.	$\boldsymbol{w} \in \mathcal{R}^N$.

LASSO

$$\|\boldsymbol{w}\|_2^2 = w_1^2 + w_2^2 + \cdots + w_N^2$$

$$\|\boldsymbol{w}\| = |w_1| + |w_2| + \cdots + |w_N|$$

目标函数是否是二阶连续可微凸函数？

$$\nabla(\|\mathbf{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2) = 2\mathbf{A}^T(\mathbf{A}\boldsymbol{w}-\boldsymbol{b})$$

$$\nabla^2(\|\mathbf{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2) = 2\mathbf{A}^T\mathbf{A}$$

$2\mathbf{A}^T\mathbf{A}$是半正定矩阵，因为它是实对称矩阵、且对任意向量$\boldsymbol{u} \in \mathcal{R}^n$有$\boldsymbol{u}^T(2\mathbf{A}^T\mathbf{A})\boldsymbol{u} = 2\|\mathbf{A}\boldsymbol{u}\|_2^2 \geq 0$

例：线性回归

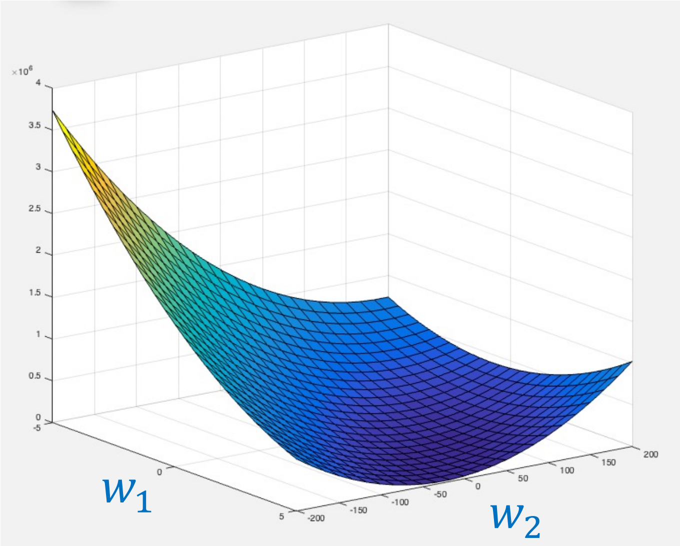
考虑$\boldsymbol{w}=(w_1,w_2)^T$的简化情况

线性回归（L2正则化）

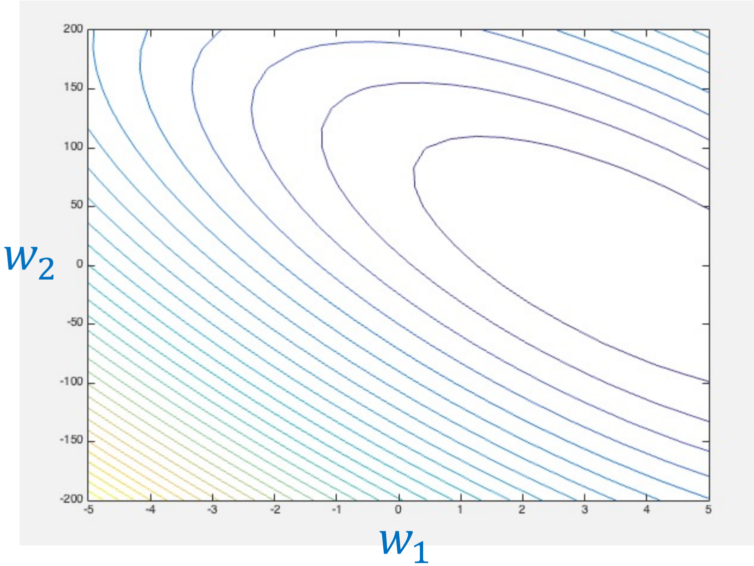
$$\min \quad \frac{1}{M}\|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2$$
$$\text{var.} \quad \boldsymbol{w}\in\mathcal{R}^N.$$

面积	室	售价
125.73	3	540
85.25	2	378
181.54	4	620

$$\|\boldsymbol{w}\|_2^2=w_1^2+w_2^2+\cdots+w_N^2$$



$\lambda=10$情况



等高线图

本讲小结

- 算法复杂度、P问题与NP问题
- 最优化问题的一般形式
- 凸集、凸函数、无约束凸优化问题

主要参考资料

Harvard CS50 <Introduction to Artificial Intelligence with Python> Slides

The OR Society《运筹学起源故事》视频

statistics.com "Safety in Numbers – Calculating Probabilities for Convoys" 网页

iq.opengenus.org "Time Complexity of Insertion Sort" 网页

David Evans CS3102 <Theory of Computation> Slides

Hackerdashery 《世界七大数学难题之一: P与NP复杂度问题》视频

Wikipedia 关于Sorting algorithm、Gamma function等的词条

Carlos Fernandez-Granda <DS-GA1013/MATH-GA2821 Optimization-based Data Analysis> Lecture Notes

谢谢！