

本笔记是从上海交通大学软件学院的计算机中的数学基础(凸优化)整理而来,还在持续更新中,文档并没有仔细校对过,很大一部分的课上的即时记录,肯定存在错误。仅供学习使用,如果侵犯任何个体或组织的权益,请联系我进行删除。

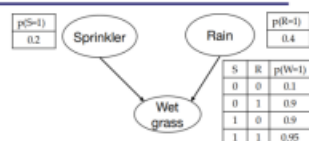
## 20210408 凸优化

贝叶斯公式:拿到一个证据以后,假设概率=先验概率\*似然/证据本身出现的概率。

概率推理:我们要去发现数据中的不同的属性,以及不同变量的一些联系。通过联合概率密度,可以回答一些我们感兴趣的问题。但是联合概率表非常庞大,会指数爆炸。我们就引入了更紧凑的一个表达,也就是贝叶斯网络。我们可以把一个很大的表转化为一些局部的信息。有了贝叶斯网络以后,就可以进行推理了。

地面潮湿可能是下雨或者洒水器打开,也就是有两个条件。

### Causal Inference (Two Causes)



- Causal or predictive inference: if the sprinkler is on, what is the probability that the grass is wet?

$$\begin{aligned}
 P(W | S) &= P(W | R, S)P(R | S) + P(W | \sim R, S)P(\sim R | S) \\
 &= P(W | R, S)P(R) + P(W | \sim R, S)P(\sim R) \\
 &= 0.95 \times 0.4 + 0.9 \times 0.6 \\
 &= 0.92
 \end{aligned}$$

我们知道草地潮湿这个证据,我们想要推断洒水器打开的概率。可以分解成,洒水器打开时,对是否下雨进行分类讨论。然后利用洒水和下雨的独立性进行化简。这时候我们发现表里对应的值都有了。可以代入计算。

### Diagnostic Inference (Two Causes)

If the grass is wet, what is the probability that the sprinkler is on?

$$P(S | W) = \frac{P(W | S)P(S)}{P(W)}$$

where

$$\begin{aligned}
 P(W) &= P(W | R, S)P(R, S) + P(W | \sim R, S)P(\sim R, S) + \\
 &\quad P(W | R, \sim S)P(R, \sim S) + P(W | \sim R, \sim S)P(\sim R, \sim S) \\
 &= P(W | R, S)P(R)P(S) + P(W | \sim R, S)P(\sim R)P(S) + \\
 &\quad P(W | R, \sim S)P(R)P(\sim S) + P(W | \sim R, \sim S)P(\sim R)P(\sim S) \\
 &= 0.52
 \end{aligned}$$

So

$$P(S | W) = \frac{0.92 \times 0.2}{0.52} = 0.35 > P(S) = 0.2$$

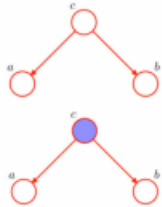
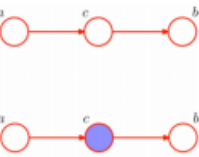
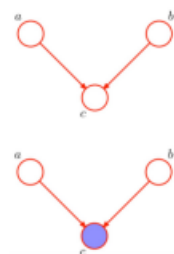
我们想知道草地潮湿的情况下,洒水器打开的概率。证据(草地潮湿这个叶子节点)本身出现的概率  $P(W)$  是没有给的,所以我们要根据顶层结点进行计算。 $W$  有两个父结点,洒水与否和下雨与否组合有四种情况,分别进行计算加和即可。

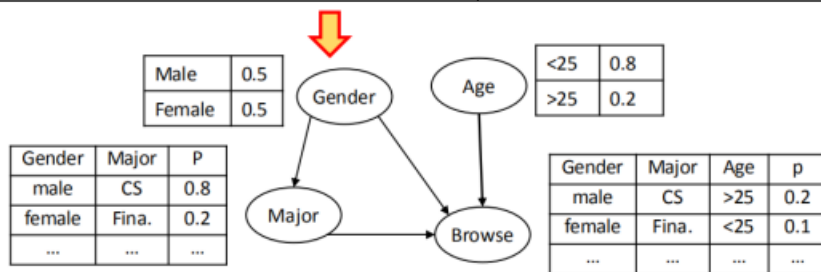
超大的贝叶斯网络该如何计算呢?如果向上一,一步步进行推理的话,等价于推理一个 3-SAT 这个 NP 问题。那么怎么办呢?我们就想到对这个贝叶斯网络进行化简,有两种方法:

1. 简化这个网络，这个贝叶斯网络虽然很大，但是变量之间有很多独立性。可以利用独立性去掉一些结点。

2. 我们不去一步步地算它，比如我们要计算  $P(S|W)$ ，我们就把  $w=1$  告诉这个网络，然后让网络进行采样。然后利用频率估计法来得到概率。

条件独立性的一些基本的结论：

 <ul style="list-style-type: none"> <li>• With <math>c</math> unobserved  <math>p(a, b, c) = p(a c)p(b c)p(c)</math>  Not conditional independence <math>a \not\perp b \mid \emptyset</math></li> <li>• With <math>c</math> observed  <math>p(a, b c) = p(a c)p(b c)</math>  Conditional independence <math>a \perp b \mid c</math></li> </ul>	<p>贝叶斯网络中如果出现了这个形状，一个结点连到了其他两个节点，如果我们没有观察到 <math>c</math>，<math>a</math> 和 <math>b</math> 是不独立的。如果我们观察到了 <math>c</math> 的结果，那么 <math>a</math> 和 <math>b</math> 就是独立的。</p>
<p><b>Head-to-Tail</b></p>  <ul style="list-style-type: none"> <li>• With <math>c</math> unobserved  <math>p(a, b, c) = p(a)p(c a)p(b c)</math>  Not conditional independence <math>a \not\perp b \mid \emptyset</math></li> <li>• With <math>c</math> observed  <math display="block">p(a, b c) = \frac{p(a, b, c)}{p(c)}</math> <math display="block">= \frac{p(a)p(c a)p(b c)}{p(c)}</math> <math display="block">= p(a c)p(b c)</math>  Conditional independence <math>a \perp b \mid c</math></li> </ul>	<p>这种形状结论是一样的，中间结点的信息会影响边上两个结点的关系。（中间这个结点的信息干扰了 <math>a</math> 到 <math>b</math> 的传递）</p>
<p><b>Head-to-Head</b></p>  <ul style="list-style-type: none"> <li>• With <math>c</math> unobserved  <math>p(a, b, c) = p(c a)p(a)p(b)</math>  Marginalize both sides over <math>c</math>  <math>p(a, b) = p(a)p(b)</math>  Conditional independence <math>a \perp b \mid \emptyset</math></li> <li>• With <math>c</math> observed  <math display="block">p(a, b c) = \frac{p(a, b, c)}{p(c)}</math> <math display="block">= \frac{p(a)p(b)p(c a, b)}{p(c)}</math>  Not conditional independence <math>a \not\perp b \mid c</math></li> </ul>	<p>这时候我们有一个相反的结论，没有观察到 <math>c</math> 的时候 <math>a</math> 和 <math>b</math> 是没有关系的。一旦观察到 <math>c</math> 了，这时候 <math>a</math> 和 <math>b</math> 就不再独立了。</p>

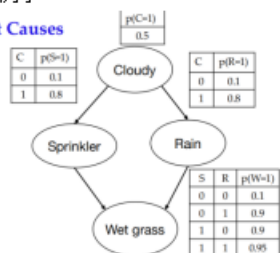


例子 1：如果  $browse$  不知道是什么时候，性别和年龄自然无关。我们知道浏览了什么时候，性别就和年龄有关系了。如果我们又知道了性别是男，那么我们就知道了年龄可能会大一点。

如果我们知道了性别是女,可能年龄就比较大。这就是通过我们已知的浏览网页的这个证据,对性别的年龄之间产生了一些关系。

例子 2:

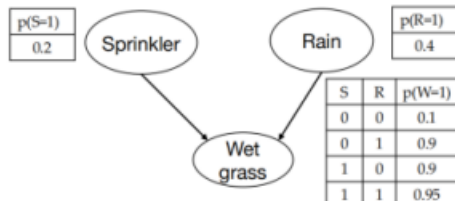
Causes



不知道 cloudy 的时候,看到洒水器打开,对于 cloudy 有新的认识。洒水器的概率影响了 cloudy 的概率,进而影响了 rain 的概率。所以就洒水和下雨就相关了。假设我们已经观察到了 cloudy,洒水器打开与否对于 cloudy 已经没用影响了,进而对 rain 就没有影响了,这时候洒水器和下雨就变成独立的了。

第二种情况是,洒水已知的情况下,cloudy 和 wet grass 是独立的。Sprinkler 不知道是時候,cloudy 的概率会影响洒水,进而影响 wet grass。但是如果已知 sprinkler 了, wet grass 就和 cloudy 无关了。

举个例子来说,我们知道 w 的时候, r 和 s 有没有影响。



因为我们知道 R, 我们把原始的贝叶斯公式每个项都加了一个 |R。

$$P(S | R, W) = \frac{P(W | R, S)P(S | R)}{P(W | R)} = \frac{P(W | R, S)P(S)}{P(W | R)} = 0.21$$

- Explaining away:

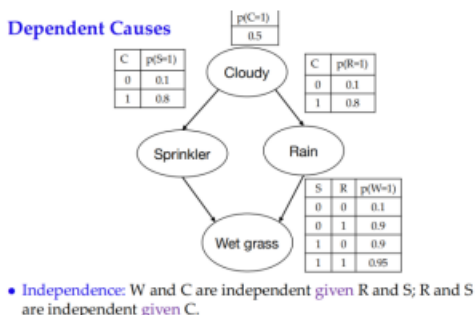
$$0.21 = P(S|R,W) < P(S|W) = 0.35$$

- Knowing that it has rained **decreases** the probability that the sprinkler is on.
- Knowing that the grass is wet, rain and sprinkler become **dependent**:

$$R \perp\!\!\!\perp S | W$$

我们发现一旦知道了  $w$  这个事件以后,  $s$  和  $r$  是独立的。为什么我地面潮湿以后, 下雨和洒水器的概率变成独立了呢? 本来就是独立的。

我们可以推广到四个结点的情况

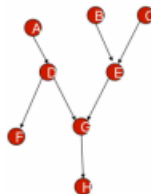


给定  $S$  和  $R$  后,  $W$  和  $C$  独立。给定  $C$  后,  $S$  和  $R$  独立。

针对, 更加一般的情形。只要拿到一个贝叶斯网络的一个子集信息, 那么这个子集和其以外的所有变量都是独立的。这个子集叫做马尔科夫毯。对于一个结点来说, 它的马尔科夫毯包括了六个结点。一旦知道了这六个结点, 那么和其他的所有结点都是独立的。这样就不需要对整个贝叶斯网络都进行一遍推理。这里有一个小题目,  $D$  结点的马尔科夫毯是什么。

What is the Markov blanket of  $D$ ?

1.  $A, E, G$
2.  $A, E, G, E$
3.  $A, E, G, E, B, C$
4. Not Sure



选择 2, 是  $AFGE$ 。

在可信的贝叶斯网络中，一个结点的马尔科夫毯即为该结点的父结点、子结点以及子结点的父结点<sup>[39]</sup>。在如图 2-2 所示的贝叶斯网络中，结点 X1、X2 通过一条有向边指向结点 T，那么 X1、X2 为结点 T 的父结点。结点 T 通过一条有向边指向结点 X6、X7，那么 X6、X7 为 T 的子结点。由于 T 与其父结点、子结点通过一条有向边相连，那么也称 T 与这些结点是邻接的。T 与结点 X3、X4、X5、X8 之间没有有向边相连，那么也称 T 与这些结点是不邻接的。结点 X8 通过一条有向边指向 T 的子结点 X7，即 T 与 X8 有共同的子结点。那么称 X8 为 T 的配偶。结点 T 的马尔科夫毯由 T 的父结点 X1、X2，T 的子结点 X6、X7，T 的配偶 X8 组成，记为  $MB(T)=\{X1, X2, X6, X7, X8\}$ 。

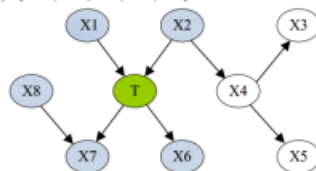


图 2-2 贝叶斯网络实例

马尔科夫毯 <https://www.cnblogs.com/wt869054461/p/9899929.html>

如果知道了 G 的信息，E 和 A 是否独立呢？

解：对于头对头的情形，已知 G，所以 D 和 E 不独立。事实上这是不一定的，对于 E，A 虽然不在 E 的马尔科夫毯中，但是 E 的马尔科夫毯中有些结点未知，所以不好说 A 和 E 到底是否独立。

刚才我们讲了贝叶斯网络如何进行简化的第一种方法，也就是马尔科夫毯利用条件独立性消掉一些结点。

第二种是基于采样

我们想查询学生喜欢的网站，我们有一些证据。

一般的方法就是在贝叶斯网络中，从根节点开始一直去推理。是否有很少的计算量去完成这件事呢？就是从网络中进行采样，获得一个后验概率的模拟。

假如说我们现在一直 S 和 R，我们想知道 C 出现的概率。

$P(\text{Cloudy} | \text{Sprinkler}=T, \text{Rain}=T)?$



- Samples of Cloudy given Sprinkler=T & Rain=T):  
1 0 1 1 0 1 1 1 0
- Posterior probability of taking on any value given some evidence:  $\Pr[Q | E_1=e_1, \dots, E_k=e_k]$ 
  - $P(\text{Cloudy} = T | \text{Sprinkler}=T, \text{Rain}=T) \approx .7$
  - $P(\text{Cloudy} = F | \text{Sprinkler}=T, \text{Rain}=T) \approx .3$

我们用 sampling 的方法来解决这个问题，我们把洒水器和下雨设成 1，我们在网络里进行采样。我们知道 S 和 R 如果都是 1 的时候，我们知道 W 的概率是多少。我们根据这个概率去掷骰子，可以拿到一些样本。这时候我们在贝叶斯网络中去采样 C。

如何得到这些采样的样本呢（如何进行采样）？在介绍具体的采样过程之前，我们介绍简单的情形。Cloudy 在这个网络中是根节点，和别的结点没什么关系。换句话说，我们如果

知道一个变量的概率，我们就按照这个概率设置随机数的接受范围。

### Sampling Single Variable



- Want to sample  $S$  when  $C = 0$
- Simple approach

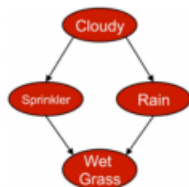
```
o r = random # generator between (0, 1)
o if (r < 0.5): sample = s+ (S=1)
o else sample = s- (S=0)
```

$P(S|C)$

C	S	p
1	1	0.9
1	0	0.1
0	1	0.5
0	0	0.5

如果我们尝试对  $S|C=0$  进行采样，同样的我们可以设置概率为 50% 的随机数。

介绍最简单的随机采样的方法：随机采样。我们在贝叶斯网络中先找一个拓扑图。在这个基础上，根据这个拓扑图的顺序依次地对图进行采样。



换句话说，先采样  $C$ ，再采样  $S$  和  $R$ ，再采样  $W$ 。

首先从  $C$  开始采样， $C$  的概率分别是 0.5, 0.5。我们生成一个随机数。发现其小于 0.5，这样  $C$  就设置为 1。在知道  $C=1$  以后，我们在  $C=1$  的基础上，对于  $S$  和  $R$  进行采样。我们按照  $S|C=1$  和  $R|C=1$  的概率进行采样。然后采样  $W|S, R$ 。最后就得到了一个样本。一旦我们拿到了这些样本，我们想得到某个概率，直接对这个样本进行计数。

刚才这个采样我们发现有点问题，每次采样都要从初始结点走一遍。我们每次采样一个样本，这个贝叶斯网络都要完整地过一遍。比较低效，还有一种方法叫做马尔科夫链蒙特卡洛法。

### Markov Chain Monte Carlo Methods

- Direct sampling generates each new sample from scratch
- MCMC generate each new sample by making a random change to preceding sample
- Can view algorithm as being in a particular state (assignment of values to each variable)

我们每次不是从  $C$  开始进行采样。我们每次都是从上次样本的基础上，进行一些调整。也就是我上一个样本是 1, 0, 1, 1。我们把 1011 的样本拿过来，从中随机地修改一些状态，这样我们就不需要每次都把一整个贝叶斯网络拿过来采样一遍。

**local variables:**  $\mathbf{N}$ , a vector of counts for each value of  $X$ , initially zero  
 $\mathbf{Z}$ , the nonevidence variables in  $bn$   
 $\mathbf{x}$ , the current state of the network, initially copied from  $\mathbf{e}$

initialize  $\mathbf{x}$  with random values for the variables in  $\mathbf{Z}$

**for**  $j = 1$  to  $N$  **do**

**for each**  $Z_i$  in  $\mathbf{Z}$  **do**

        set the value of  $Z_i$  in  $\mathbf{x}$  by sampling from  $P(Z_i | mb(Z_i))$

$N[x] \leftarrow N[x] + 1$  where  $x$  is the value of  $X$  in  $\mathbf{x}$

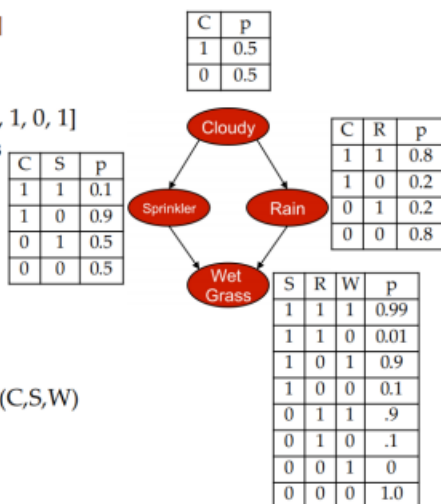
**return**  $NORMALIZE(\mathbf{N})$

状态向量记录当前样本的状态。根据马尔科夫毯的其他变量的特征，来计算这个概率。

## Gibbs Sampling: Example



- Want  $\Pr(R|S=1, W=1)$  //  $[X=R, e=[S, W]]$
- Non-evidence variables  $\mathbf{Z} = \{C, R\}$
- Initialize randomly:  $C = 1$  and  $R = 0$
- So the initial state  $\mathbf{x} = (C, S, R, W) = [1, 1, 0, 1]$
- Sample  $C$  given current values of its Markov Blanket ( $S$  &  $R$ )  
i.e.,  $C \sim P(C|S=1, R=0)$
- First have to compute  $P(C|S=1, R=0)$
- Use exact inference to do this
- Sample  $C$  given  $P(C|S=1, R=0)$
- Get  $C = 0$
- New state  $\mathbf{x} = (0, 1, 0, 1)$
- Sample  $R$  given its Markov Blanket ( $C, S, W$ )
- Suppose result is  $R=1$
- New state  $\mathbf{x} = (0, 1, 1, 1)$
- ...



我们知道了  $S$  和  $W$ ，目标是求  $\Pr(R|S=1, W=1)$ 。

此时未知量是  $C$  和  $R$ ，放到集合  $\mathbf{Z}$  里。已知量是  $S=1, W=1$ 。

因为  $C$  和  $R$  未知，任取一个值。这时候我们就拿到了一个初始样本  $(C, S, R, W) = [1, 1, 0, 1]$ 。

我们要在初始样本的基础上去调整一个  $C$ ，因为  $C$  的初始值只是一个随机数。我们在  $1, 1, 0, 1$  的基础上对  $C$  进行修正。我们要求  $C$  的概率，我们可以通过  $C$  的马尔科夫毯来计算  $C$  的后验概率。即算了  $P(C|S=1, R=0)$ ，算出这个概率以后，根据这个概率去采样  $C$ 。假如我们采样的  $C$  是  $0$ ，我们把  $1101$  修改为新的样本  $0101$ ，这样就完成了一次变量转移。然后我们继续通过  $R$  的马尔科夫毯计算  $R$  的修正概率，然后按照  $R$  的修正后验概率进行采样。生成新的样本。

为什么要这样采样？为什么这样采样以后就能拿到后验概率呢？这里面涉及到一个马尔科夫链的收敛性的原理，我们把采样到的样本作为一个状态，状态和状态之间会发生转移。采

样  $t$  步的话，状态就转移了  $t$  次，最后就会生成一个状态转移图。我们采样就是在这个状态转移图中按照某种概率调到下一个样本。这个图最后有没有可能达到一种稳定状态呢？我们把这个转移的过程写成一个矩阵的形式。

$$\begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix} \begin{bmatrix} P11 & & & \\ & & & \\ & & & \\ & & & P44 \end{bmatrix}$$

每次做一个转移就等于右乘一个矩阵，反复乘以后会收敛到一个

稳定状态  $\pi$ 。这个吉布斯采样可以证明最终收敛到一个稳定的概率分布，无论如何转移都不变。

非周期的马尔科夫链，且其任意两个状态是连通的，那么会存在平稳分布（收敛性）

吉布斯采样本质上是构造这样的一个马尔科夫链，因此理论必然是收敛的。

如果实验发现不能收敛，可能是计算的时间不够长或者设置的主题数太大了。

**定理0.4.2** 如果一个非周期马氏链具有转移概率矩阵  $P$ ，且它的任何两个状态是连通的，那么  $\lim_{n \rightarrow \infty} P_{ij}^n$  存在且与  $i$  无关，记  $\lim_{n \rightarrow \infty} P_{ij}^n = \pi(j)$ ，我们有

$$1. \lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

$$2. \pi(j) = \sum_{i=0}^{\infty} \pi(i) P_{ij}$$

3.  $\pi$  是方程  $\pi P = \pi$  的唯一非负解

其中，

$$\pi = [\pi(1), \pi(2), \cdots, \pi(j), \cdots], \quad \sum_{i=0}^{\infty} \pi_i = 1$$

$\pi$  称为马氏链的平稳分布。

应用:疾病症状推理、文本分类。

## 2021/4/12 第二节课 信息

在通讯的这些人，关注的就是信息如何更好地传播。

如何度量信息呢？

比如一个小孩只会睡觉和哭闹，为了传达这两个状态，我们可以用 1 个比特来表示。随着孩子的长大，小孩除了睡觉和哭闹，还会调皮和微笑，此时这四个状态可以用 2 个比特来表示。

再长大以后，如果有  $M$  个状态，我们可以用  $\log_2(M)$  位的存储来记录这个消息。假如一天



中出现了  $N$  次  $M$  个状态的事件，那么对这个的量化就是  $N \log_2(M)$  bit。如果我们想把一个密码传递给别人，这个字符串有  $N$  个字符，每个字符有 26 个状态，类似地，这个消息的传递就需要  $N \log_2(26)$  bits。一张  $512 \times 512$  的灰度图，有  $512 \times 512 \times \log_2(256) = 262144$  字节。

比如一天中观察了一个小孩五次，都是睡觉。假如晚上七点第六次观察还是在睡觉，那么没有观察到什么新的东西。如果半夜观察到这个小孩突然哭起来了，这个事件传达的信息更多，因为我们之前从没想到。我们认为小孩的正常状态就是睡觉。睡觉和哭泣为什么有区别呢？因为睡觉是之前观测到的事件的重复，而哭泣这个事件是稀有的事件，使得原先不确定的事情清楚了。一个事件的多与少取决于不确定性，对于一个低概率事件，如果出现了就会给我们传递更多的信息。而一个高概率的，都知道会发生的事件，传递给我们的信息就会比较少。香农：“信息是关于不确定性的消息。”

信息和概率的关系？信息和事件的概率有一个度量关系，一个概率为  $p$  的事件发生了，

信息就是  $\log_2 \frac{1}{p}$  bits。

Information gained upon learning  
event of probability  $p$  occurred

假如我们有一个事件是抛硬币，就各自有一个 bit 的信息。

#### Example 1: coin tossing

-  $x = [\text{heads}; \text{tails}]$ ,  $p = [1/2; 1/2]$ ,  $\text{SIC} = [1; 1]$  bits

明天是不是生日？如果明天不是我的生日，这个消息对我来说没有太大的惊喜，信息量只有 0.004。如果明天是我的生日，这个消息就比较有价值，信息就比较多。

#### Example 2: is it my birthday?

-  $x = [\text{no}; \text{yes}]$ ,  $p = [364/365; 1/365]$ ,  $\text{SIC} = [0.004; 8.512]$  bits

#### Information must

1. be **something**, although the exact nature (substance, energy, or abstract concept) isn't clear;
2. provide **"new"** information: a repetition of previously received messages isn't informative;
3. be **"true"**: a lie or false or counterfactual information is *mis-information*, not information itself;
4. be **"about"** something.

$$S = K \log W$$

$K$  – Boltzmann's constant

$W$  – the number of microscopic states or configurations.  
(e.g., different hues and ways that atoms can take)

在微观状态下，信息可以用微观分子的状态数来度量，概念其实是统一的。熵就是一个期望的信息或者是一个平均的信息。对于一个随机变量  $x$ ，如果有  $M$  个取值，对于每个取值的概率分别为  $p_1$  到  $p_M$ ，对于这个随机变量的熵如下计算：

$$H(X) = \mathbb{E} \left\{ \log \frac{1}{p(X)} \right\} = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \text{ bits}$$

从这个公式可以看出来，这个熵定义了一个概率分布的总体的不确定性。熵越大，总体的不

确定性就增大，对于这个结果的置信度就会越小，因为我们越来越不确定它。  
熵的推论：

- $H(X)$  = the **average** Shannon Information Content of  $X$
- $H(X)$  = the **average** number of bits required to represent or transmit an event drawn from the probability distribution for  $X$
- $H(X)$  = the **average** information gained by knowing its value
- $H(X)$  = the **average** number of "yes-no" questions need to find  $x$  is in the range  $[H(x), H(x)+1)$

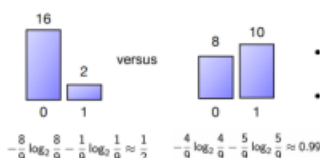
1. 熵就是信息的一种度量。
2. 如果把  $x$  认为是服从一个概率分布的事件，如果我们要表示这个事件，平均下来我们需要多少个比特。
3. 熵还可以表示知道这样一个事件的结果以后，获取的信息增益是多少。
4. 我们想知道  $x$  的数值，我们关于  $x$  去问一些 yes-no 问题，eg: 问  $x$  是否满足哪个特征。平均要问几个问题，才能确定出  $x$  的值。
5. 熵是消解信息不确定性，所平均需要付出多少代价。

Eg: 分别抛两个硬币

#### We Flip Two Different Coins

Sequence 1:  
00010000000000100 ... ?

Sequence 2:  
010101110100110101 ... ?



- Entropy measures the expected "surprise"

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x)$$

- Entropy measures the information content of each observation
- A fair coin flip has 1 bit of entropy

一个不均匀的硬币的熵更少，一个均匀的硬币的熵更大。说明平均意义上，一个均匀的硬币给我们更大的惊喜。

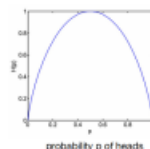
针对抛硬币问题，我们可以泛化到一般情况：

#### (1) Bernoulli Random Variable (e.g., Flipping Coin)

$$x = \{0, 1\}, p_x = [1-p, p]$$

$$H(p) = -(1-p) \log(1-p) - p \log p$$

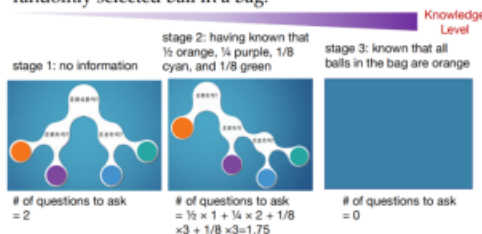
\* we often write  $H(p)$  to mean  $H([1-p, p])$ .



我们可以看到在伯努利问题中，熵在  $p=0.5$  时取到最大。

熵是信息不确定性的度量，如果熵不变的话就代表没有获得新的信息。

Ask as few questions as possible to verify the color of a randomly selected ball in a bag.



Entropy:

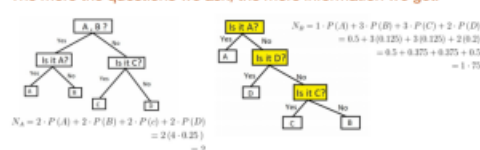
- 考虑到上图这个游戏，从左到右三种分别是玩家猜袋子中的这个球的游戏。
- 最左边玩家什么信息都不知道，所有平均的二分询问查找即可，平均 yes-or-no 的询问次数为 2 次。
  - 中间情况，玩家已经知道了一些球的颜色比例，玩家可以设置一个更好的策略，使得平均 yes-or-no 的询问次数降低。
  - 第三种玩家已经完全知道了袋子中球的信息（全是橙色的球），那么玩家不需要再问问题，直接拿就行。
- 我们可以看到从左到右，玩家所知道的信息在增加、对信息的确定度在上升、信息的熵在下降。与熵的第四个推论相同。

Which machine produces more information?

A	A B D C C A B D D C	$P(A) = 0.25$	$P(A) = 0.5$
B	A C B D A A B C B D	$P(B) = 0.25$	$P(B) = 0.125$
		$P(C) = 0.25$	$P(C) = 0.125$
		$P(D) = 0.25$	$P(D) = 0.25$

Probabilities for Machine A      Probabilities for Machine B

The more the questions we ask, the more information we get!



例子 2: 机器按照一定概率生成字符，我们可以用 yes-no 问题的平均个数来衡量机器产生字符串的不确定度。我们发现系统生成的状态越均衡，熵越大。

我们从数学中推广到日常生活中。在热力学中也有一个熵的概念（微观粒子状态的总和），这两个概念是一回事，本质上都是信息。

联合熵就是考虑几个变量联合的一个熵，这里我们考虑随机变量  $X$  和  $Y$ 。

- Extend the notion to a pair of discrete random variables ( $X, Y$ )

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y)$$

$$H(X, Y) = -E \log p(X, Y)$$

对每个状态我们也去计算  $-\log p(x, y)$  的期望

## Conditional Entropy



Entropy of a random variable given another random variable.

If  $(X, Y) \sim p(x, y)$

$$\begin{aligned} H(Y|X) &= \sum_{x \in X} p(x) H(Y|X=x) \\ &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y|x) \end{aligned}$$

Conditional entropy is the expected value of specific conditional entropy  $E_{p(x,y)}[H(Y|X=x)]$

- $H(Y|X) \neq H(X|Y)$

条件熵的计算，对于  $x$  的每种取值  $x$ ，计算一个  $H(Y|X=x)$  熵，最后加权平均。

Example:  $X = \{\text{Raining, Not raining}\}$ ,  $Y = \{\text{Cloudy, Not cloudy}\}$

$P(X, Y)$	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

What is the entropy of cloudiness  $Y$ , given that it is raining?

$$\begin{aligned} H(Y|X=x) &= - \sum_{y \in Y} p(y|x) \log_2 p(y|x) \\ &= - \frac{24}{25} \log_2 \frac{24}{25} - \frac{1}{25} \log_2 \frac{1}{25} \\ &\approx 0.24 \text{ bits} \end{aligned}$$

条件熵计算的例子： We used:  $p(y|x) = p(x, y)/p(x)$ , and  $p(x) = \sum_y p(x, y)$

条件熵是观察到一个变量以后，对其他变量获得的额外的信息。

$H(Y|X)$  is the average **additional information** in  $Y$  when you know  $X$



Take a **weighted average** of the entropy of each row using  $p(x)$  as weight

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X=x) = \frac{1}{4} H(\text{cloudy}|\text{is raining}) + \frac{3}{4} H(\text{cloudy}|\text{not raining})$$

### Probabilities

$$P(X, Y, Z) = P(Z|X, Y)P(Y|X)P(X)$$

### Entropy

$$H(X, Y, Z) = H(Z|X, Y) + H(Y|X) + H(X)$$

$$H(X_{1:n}) = \sum_{i=1}^n H(X_i|X_{1:i-1})$$

熵中虽然也存在链式规则，但是其之间是加法

交叉熵

在机器学习中非常多，用来判断一个机器学习方法的好坏。假如一个变量的真实分布是  $P$ ，eg 一个球的真实分布是  $1/2, 1/4, 1/8, 1/8$ 。我设计了一个方法，猜这个球的分布是  $Q$ ： $1/4, 1/4, 1/4, 1/4$ 。这时候我们会多少代价。

## Cross Entropy



The cost of resolving uncertainty using a distribution  $q$  given that the real distribution is  $p$ .

$$CE(p, q) = \mathbb{E}_{x \sim p(x)} \left\{ \log \frac{1}{q(x)} \right\} = - \sum_{x \in X} p(x) \log_2 q(x)$$

**Example:** recall the ball color verification game: what if still using the strategy for stage 1  $q = (1/4, 1/4, 1/4, 1/4)$  even the knowledge level is in stage 2 (i.e., having already known that the real distribution  $p = (1/2, 1/4, 1/8, 1/8)$ )?

Now the # of questions to ask is  $1/2 \times 2 + 1/4 \times 2 + 1/8 \times 2 + 1/8 \times 2 = 2 > 1.75$



Cross Entropy is widely used in **Machine Learning** to measure the quality of estimated distribution  $q$  with respect to the real data distribution  $p$ .

换句话说，就是在真实分布  $p$  下，回答  $q$  这个分布的  $-\log$  期望。

2021/4/15

联合熵：根据联合概率求一个期望。

条件熵：在已知一个变量的条件下，期望等于多少。

条件熵有两种解释：1. 在条件概率列表里，只取一行（换句话说，固定了  $x$  为某个常数），关注某个变量的熵。即可求得  $H(Y|X=0)$  和  $H(Y|X=1)$ ，我们再按照  $x$  出现的概率进行加权，即可求得  $H(Y|X)$

我们也可以理解为，当你知道  $x$  以后， $H(Y|X)$  是平均额外信息增量。

后来，我们又讲到交叉熵这个概念。换句话说数据真实的分布是  $p$ ，另一个分布是  $q$ 。在真实分布条件下， $q$  这个分布有多大的不确定度。我们回到猜球游戏。假如我们采用平均问问题的情况，那么平均需要 2 个问题。如果我们已知了球在袋子中概率分布，我们按照球的概率分布问问题，那么平均需要 1.75 个问题。（橙色 1 个问题、紫色 2 个问题、蓝色和绿色 3 个问题）

如果在已知球概率分布的情况下，依旧使用平均策略，会怎么样呢？平均策略下，四种颜色的球都要 2 个问题。

$$CE(p, q) = -(\log_2(0.25) \times (0.5 + 0.25 + 0.125 + 0.125)) = 2$$

$$\begin{aligned} H(p) = CE(p, p) &= -(\log_2(0.5) \times 0.5 + \log_2(0.25) \times 0.25 + 2 \times \log_2(0.125) \times 0.125) \\ &= -(-1 \times 0.5 - 2 \times 0.25 - 2 \times 3 \times 0.125) = 1.75 \end{aligned}$$

$$CE(p, q) - H(p) = 0.25$$

对数据的无知导致代价更大。

## 相对熵

和真实的到底相差多少，需要相对熵的概念。

## Measure of distance between two distributions $p$ and $q$

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

- Also known as **Kullback-Leibler** divergence in statistics: expected log-likelihood ratio.
- Measures the “distance” between the probability distributions  $p$  and  $q$ .
- A measure of inefficiency of assuming that distribution is  $q$  when the true distribution is  $p$ .
- If we use distribution  $q$  to construct code, we need  $H(p) + D(p||q)$  bits on average to describe the random variable.

换句话说，数据存在一个真实分布，我们预测的模型存在一个分布。我们对每个连续点上对  $P(x)-Q(x)/P(x)$  求一个期望的比值。

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = CE(p, p) - CE(p, q) = H(p) - CE(p, q)$$

互信息：两个变量  $x$  和  $y$ ，共同拥有的一部分信息。换句话说，知道  $x$  以后，就会有互信息这些不确定度被消除。

- Measure of the amount of information that one random variable contains about another random variable

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = D(p(x, y)||p(x)p(y))$$

- The average amount of information that you get about  $x$  from observing the value of  $y$ .
- Reduction in the uncertainty of one random variable due to the knowledge of the other

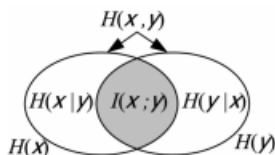
关于  $xy$  联合概率分布和  $x$ 、 $y$  边缘概率分布的乘积的距离。可以理解为真实的分布是  $p(x,y)$ ，现在的分布是  $p(x)p(y)$ 。等价于知道一个信息以后，另一个变量的不确定度降低了多少。

- Relationship between entropy and mutual information

$$I(X;Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y)$$

Information in x      Information in x when you already know y

- $I(X;Y)$  is the intersection of information in X with information in Y



- $I(X;Y) = I(Y;X)$

图 1 很清楚的一张图，建议仔细看看

上图描述了熵、条件熵、联合熵、互信息的关系。

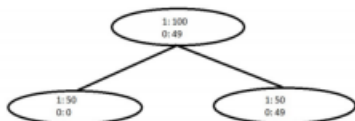
决策树设计了一些规则，对数据进行划分，比如宽度是否大于 6.5cm。

问题 1：为什么用划分的思想？

这其实就是一个熵（不确定度）逐渐消解的一个概念。比如说，我们有一堆橙子 and 柠檬。一开始我们对它们的信息一无所知。然后在某个维度进行了一次划分，我们发现这两半的类别，更加清楚了，熵减小了。继续划分以后，划分出来的样本的熵就更小了。这个过程关键的就是维度的选取，以及在维度上进行划分的位置。

决策树的划分准则：这样划分使得熵减少的最多。我们最终的目标就是熵为 0，最终就不需要问问题了。

What is the information gain of this split?



- Root entropy:  $H(Y) = -\frac{49}{149} \log_2(\frac{49}{149}) - \frac{100}{149} \log_2(\frac{100}{149}) \approx 0.91$
- Leafs entropy:  $H(Y|left) = 0$ ,  $H(Y|right) \approx 1$ .
- IG(split)  $\approx 0.91 - (\frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1) \approx 0.24 > 0$

计算 IG（信息增益）

## 凸优化

要找到最好的函数拟合这些样本，就是最小化均方误差。

最短路问题：

$$x_e = \begin{cases} 1 & e \in P \\ 0 & e \notin P \end{cases}$$

### Example: Shortest Path

Given a directed network  $G = (V, E)$ , find the **shortest** path from  $s$  to  $t$ .

$x_e$ : edge  $e$  is selected



$$\begin{aligned} & \text{minimize} && \sum_{e \in E} x_e \\ & \text{subject to} && \sum_{e \leftarrow v} x_e = \sum_{e \rightarrow v} x_e, \quad \text{for } v \in V \setminus \{s, t\}. \\ & && \sum_{e \leftarrow s} x_e = 1 \\ & && x_e \leq 1, \quad \text{for } e \in E. \\ & && x_e \geq 0, \quad \text{for } e \in E. \end{aligned}$$

路径的条件：从  $s$  出去的边至少有一条边

从顶点进来和从顶点出去的边应当相等，要么 0（顶点未被访问），要么 1（顶点被访问到了）。最后一个条件是  $x_e \in \{0, 1\}$ ，但是如果取整数的话，在优化中是比较复杂的情况，上述式子中采取了放松的方法。

### 例子 2：网络流问题

每条路有代价，并且也有对应的容量

### Example: Minimum Cost Flow (网络流)

Given a directed network  $G = (V, E)$  with cost  $c_e \in \mathbb{R}^+$  per unit of traffic on edge  $e$ , and capacity  $d_e$ , find the **minimum cost** routing of  $r$  divisible units of traffic from  $s$  to  $t$ .



$$\begin{aligned} & \text{minimize} && \sum_{e \in E} c_e x_e \\ & \text{subject to} && \sum_{e \leftarrow v} x_e = \sum_{e \rightarrow v} x_e, \quad \text{for } v \in V \setminus \{s, t\}. \\ & && \sum_{e \leftarrow s} x_e = r \\ & && x_e \leq d_e, \quad \text{for } e \in E. \\ & && x_e \geq 0, \quad \text{for } e \in E. \end{aligned}$$

从起始节点出去的货物综合一定是  $r$ 。对于中间结点，多少进来就必须多少出去。

Eg3:

希望设计机器人的路径，



## Example: robot trajectory planning

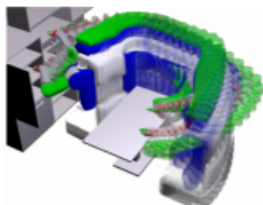


Figure from (Schulman et al., 2013)

Given robot state  $x_t$  and control inputs  $u_t$

$$\begin{aligned} & \underset{x_1:T, u_1:T-1}{\text{minimize}} && \sum_{i=1}^{T-1} \|x_t - x_{t+1}\|_2^2 + \|u_t\|_2^2 \\ & \text{subject to} && x_{t+1} = f_{\text{dynamics}}(x_t, u_t), \text{ (robot dynamics)} \\ & && f_{\text{collision}}(x_t) \geq 0.1 \text{ (avoid collisions)} \\ & && x_1 = x_{\text{init}}, x_T = x_{\text{goal}} \end{aligned}$$

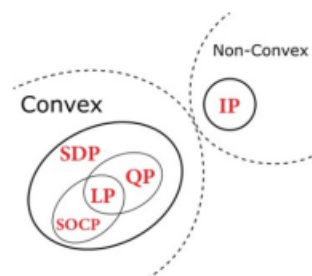
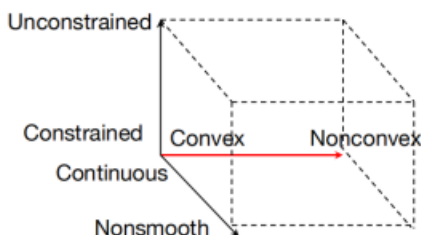
1.希望路径最短 2.希望控制信号最简单

我们使用  $x$  表示机器人的位置。我们  $t$  时刻对机器人有一个输入信号叫做  $u_t$ 。

最终目标：1.机器人走的总的路线越小越好（可能为了耗电最少）2.机器人本身的控制逻辑要最简单。我们还希望  $x_t$  这个位置，和周围物体的距离不要小于 0.1（避免碰撞）。

我们讲的这三个例子，有的是离散的、有的包含了函数的估计，但是都有相近的形式，都是在一个函数上找一个极值点。比如最小二乘/线性回归，要优化的函数就是使损失最小。第二个问题的目标函数，就是使得路径/网络流中的代价最小。第三种就是一个复杂的函数。这类问题统称为优化问题，对应的函数就叫做目标函数。

## Overview of Optimization Problems



Our focus in this course:

— Constrained, Continuous, Convex

最优化问题的分类，我们关注的三类问题

关注有约束的、连续的、凸的函数

在讲线性回归之前要复习一下线性代数的概念。

向量的表示、向量的加减法、向量的数乘。

范数：范数是作用在向量上的一个函数，可以简单的理解为向量的一个长度。

For a vector  $x \in \mathbb{R}^n$  with elements  $x = (x_1, x_2, \dots, x_n)$ :

- The  $l_2$  norm, or Euclidean norm:

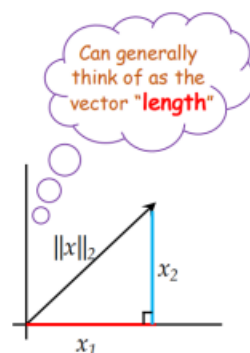
$$\|x\|_2 = \|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

- The  $l_1$  norm:

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

- The  $l_p$ -norm:

$$\|x\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + \dots + |x_n|^p}$$



2021/4/22

半正定矩阵。如果对任何向量，进行二次乘法，最终恒大于等于 0，就叫做半正定矩阵。它在几何上有一些特性。

**Example:**

- The identity matrix  $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  is positive-definite

$$z^T I z = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^2 + y^2 \geq 0$$



- A symmetric matrix  $A$  is positive definite(PD) if for all nonzero  $x \in \mathbb{R}^n$ ,  $x^T A x > 0$

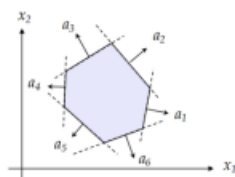
Notation:  $A \succeq 0$  if  $A$  is PSD,  $A \succ 0$  if  $A$  is PD

正定矩阵：乘积恒大于 0。

不等式约束定义了一个半平面。

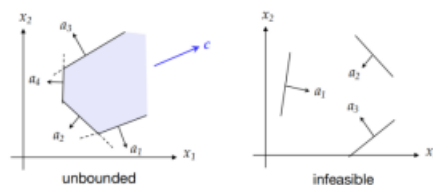
如果有  $m$  个半平面，最后的形状就是一个多边形。

Multiple halfspace constraints,  $a_i^T x \leq b_i, i = 1, \dots, m$  (or equivalently  $Ax \leq b$ ), define what is called a **polytope**.



如何证明多边形任取一点如何知道是顶点呢？如果存在以这个点为终点的线段，其无论怎么旋转，两端都在多边形中，那么这个点就不是顶点。

An LP either has an optimal solution, or is **unbounded** or **infeasible**.



Example: suppose we have both the constraints  $x_1 \geq 5$  and  $x_1 \leq 4$

单纯性法和椭球法是专门为线性规划设计的。

单纯形法：任取一个顶点，往两边移动，看哪里使得等高线降低。

线性规划的另一个思路：找约束条件的一个比例，对约束条件进行线性组合，构造目标函数的上界。问题变成最小化这样一个上界。

$$\sum_{j=1}^n a_{ij} x_j \leq b_i$$

$$\text{约束乘上线性组合系数 } y_i \sum_{j=1}^n a_{ij} x_j \leq y_i b_i$$

$$\begin{aligned} \text{subject to } & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ & \dots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_n \\ \text{and} & \\ & x_i \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned}$$

$$\text{则有 } z^* \leq \sum y_i \bar{a}_i^T \bar{x} \leq \sum b_i y_i$$

$$\sum_{i=1}^n (a_{ij}y_i)x_j \leq \sum_{i=1}^n y_i b_i$$

要保证组合以后的式子是原问题的上界  $z^* \leq \sum_{i=1}^n (a_{ij}y_i)x_j$

即要保证  $\sum_{j=1}^n (a_{ij}y_i) \geq c_i$

且  $y_i \geq 0$

这个问题就叫做对偶问题。

2021/4/26

对偶问题就是对原问题的约束条件做一个线性组合，构成所求函数的上界，我们的目标就变成了最小化这个上界。此时  $y$  叫做对偶变量，控制着原问题的约束条件所占的比例。

- $n$  products,  $m$  raw materials
- Every unit of product  $j$  uses  $a_{ij}$  units of raw material  $i$
- There are  $b_i$  units of material  $i$  available
- Product  $j$  yields profit  $c_j$  per unit
- Facility wants to maximize profit subject to available raw materials.

	$x_1$	$x_2$	$x_3$	$x_4$	
$y_1$	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$b_1$
$y_2$	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$b_2$
$y_3$	$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$b_3$
	$c_1$	$c_2$	$c_3$	$c_4$	

经济学解释：

原问题：我们回到工厂生产产品的例子，每个产品对每种原材料都有需求，并有对应的利润，求利润最大。

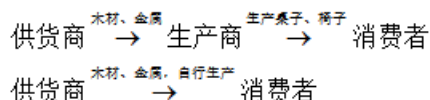
Primal LP	Dual LP
$\max \sum_{j=1}^n c_j x_j$ s.t. $\sum_{j=1}^n a_{ij} x_j \leq b_i, \text{ for } i \in [m]$ $x_j \geq 0, \text{ for } j \in [n]$	$\min \sum_{i=1}^m b_i y_i$ s.t. $\sum_{i=1}^m a_{ij} y_i \geq c_j, \text{ for } j \in [n]$ $y_i \geq 0, \text{ for } i \in [m]$

	$x_1$	$x_2$	$x_3$	$x_4$	
$y_1$	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$b_1$
$y_2$	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$b_2$
$y_3$	$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$b_3$
	$c_1$	$c_2$	$c_3$	$c_4$	

- Dual variable  $y_i$  is a proposed **price** per unit of raw material  $i$
- Dual price vector is feasible if facility has incentive to sell materials
- Buyer wants to spend as little as possible to buy materials

对偶问题：对偶问题是引入了一个  $y$ ， $y$  代表某个原材料的价格（eg：木头的价格、金属的价格） $a_1$  代表桌子需要金属的总量， $y_1 \times a_1$  代表生产桌子所需要金属的总量。

$y_1 \times b_1$  是如果把原材料全部用完，要花多少钱。



对于供货商来说，比如木材和金属基本上不值钱 ( $y^T b = 0.01$ )，生产桌椅是暴利 ( $c^T x = 100$ )，它就不会卖原料给生产商，而是自己生产桌椅出售。如果我们要保证让供货商走第一条路，那么我们必须要保证  $y^T b \geq c^T x$ ，这就是经济学的解释。

我花多少钱买原材料，这个价格要大于产品生产完的利润。是站在原材料供应商的角度，卖出这么多原材料获得的利润，要大于生产产品以后的利润，否则我就直接生产产品，不卖原材料了。在保证原材料供应商有利润的情况下，保证我购买全部原材料的花费越小越好。

物理学解释：在  $c$  这个方向施加这么大的力，这个球就会向力的方向去移动，直到被约束条件组成的墙阻挡住，这时候就达到了最优解。 $y_i \times a_i$  可以视作墙对球的反作用力，如果球是静止状态  $c^T = \sum_i y_i \times a_i$ 。如果我们希望把球推回原点，我们就发现  $c^T < \sum_i y_i \times a_i$ ，这时候反作用力就是其上界。球从原点走到最优点，把球送回原点需要最大的功就是  $\sum_i y_i \times a_i$ ，

送到最优点所做的功就是  $c^T x$ 。

还有一个很有意思的现象，球在最优点的时候，大部分约束条件都没有用。我们说线性规划的最优条件是一个多边形，我们在物理学角度来看，到最优点的时候只有两个挨着的墙和它有作用。我们还发现，对球起作用这两个墙符合球刚好球在这墙的端点上。分两种情况，

1. 墙对球没有作用力， $y_i = 0$
2. 墙对球有作用力， $y_i \neq 0$ 。约束条件刚好是满足的（tight 的情况），刚好使得  $a_i x = b$ 。

我们综合这两个条件，发现一个等式是成立的，即  $y_i (a_i x - b) = 0$ 。这个等式叫做**互补松弛条件**，我们可以用这个条件来判断是否达到了最优解。

## 弱对偶

原问题的可行解  $x$  和对偶问题的可行解  $y$ ，对于任意的  $x$  和  $y$ ，原问题的目标函数的值是小于等于对偶问题的目标函数的值。 $b^T y \geq c^T x$ 。这个有什么用呢？

第一个解释是经济学解释： $b^T y$  是所有原材料的售价， $c^T x$  是产品的利润。则卖原材料获得的收入是大于等于原材料做成产品以后的利润。

第二个是 Upper bound 解释

第三个是物理学解释,把球从原点移到最优点所做的功是大于把球从最优点移到原点所做的功。

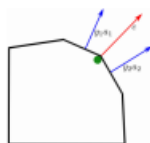
## 强对偶

在某些情况下,原问题和对偶问题有有界可行解的情况,原问题的最优解刚好等于对偶问题的最优解。也就是说在线性规划中,我们对约束条件做线性变化得到对偶问题,对偶问题的最优解刚好能落到原问题的最优解上,即可以证明  $y^* = x^*$ 。这个定理给了我们很多的帮助,

我们本来认为对偶问题只是提供了一个原问题的上界,并不知道上界的最小值是多少。但是通过这个定理我们发现求解对偶问题就能得到原问题的最优解。

我们可以在生产中去解释,供货商希望卖原材料获得利润,生产商可以不断地和他压价,直到售出材料的利润刚好等于卖出产品的利润。这两个收益对供货商来说是一样的。

第三个是物理学解释,把球移动到最优点刚好等于反作用力把球从最优点移动到原点的功,不记能量损失的话,两者的功是相同的。



Recall the physical interpretation of duality

When ball is stationary at  $x$ , we expect force  $c$  to be neutralized only by constraints that are tight. i.e. force multipliers  $y \geq 0$  s.t.

- $y^T A = c$
- $y_i (b_i - a_i x) = 0$
- $y^T b - c^T x = y^T b - y^T A x = \sum y_i (b_i - a_i x) = 0$

我们给出一个不严谨的证明,在最优解的时候,有这个表达式,移回原点的功  $y^T b$  减

去原点移动到最优点的功  $c^T x$ ,我们代入最优解时候的条件,我们发现两者之差正好等于 0。

(没有能量损失)

## 凸优化简介

线性规划是所有优化问题里面非常特殊的一个子问题,我们的重点是为大家介绍一个计算机中非常普遍的凸优化问题。优化问题的本质是在可行的方案里面,选择一些最佳的配置。要最小化/最大化某个满足一些约束的函数。我们比较关注连续型的优化,因为离散的优化通常比较复杂。连续型优化就是把约束条件写成连续型的函数,如下是连续型优化问题的标准形式。

### Finding the minimizer of a function subject to constraints:

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

where

$x = (x_1, \dots, x_n)$  is the optimization variable

$f_0: \mathbf{R}^n \rightarrow \mathbf{R}$  is the objective function

$f_i: \mathbf{R}^n \rightarrow \mathbf{R}, \quad i = 1, \dots, m$  are inequality constraint functions

$h_i: \mathbf{R}^n \rightarrow \mathbf{R}, \quad i = 1, \dots, p$  are equality constraint functions.

- **Goal:** find an optimal solution  $x^*$  that minimizes  $f_0$  while satisfying all the constraints.

有  $m$  个不等式约束  $f_i$ , 有  $p$  个等式约束函数,  $F_0$  是目标函数。目标就是在不等式约束和等式约束组成的可行域中, 找到一个使得目标函数最小化的最优解。对于连续型优化有一个很重要的问题就是局部最小的问题, 这种问题比较复杂。有没有一种函数, 它的极小值就是它的最小值, 这一类函数就是我们要介绍的凸函数 (convex function)。我们期望这个函数全局最优和局部最优相同。

#### Convex Optimization

A problem of minimizing a **convex function** (or maximizing a concave function) over a **convex set**.

Standard Form:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & g_i(x) \leq 0, \text{ for } i \in C_1 \\ & h_i(x) = 0, \text{ for } i \in C_2. \end{array}$$

where  $f_0, g_i, h_i$  are convex

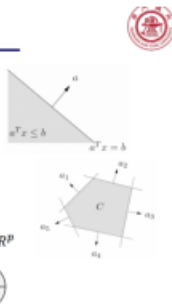
Convex optimization problems have local minima == global minima

所有这些目标函数和可行域的约束条件必须是凸函数, 这个问题就是一个凸优化问题。

凸集从字面上来讲就是一个边界都凸在外面的一个集合。非凸的证明: 一定可以找到一条线段, 线段中的某个点在图形外面。

### Examples of Convex Set

- Trivial:  $\emptyset$ , point, line, etc.
- Hyperplane:  
 $C = \{x \mid a^T x = b\}$  where  $a \in \mathbf{R}^n, b \in \mathbf{R}$
- Halfplane:  
 $C = \{x \mid a^T x \leq b\}$
- Polyhedron:  
 $C = \{x \mid Ax \leq b, Cx = d\}$   
where  $A \in \mathbf{R}^{m \times n}, b \in \mathbf{R}^m, C \in \mathbf{R}^{p \times n}, d \in \mathbf{R}^p$
- Euclidean ball:  
 $B(x_c, r) = \{x \mid \|x - x_c\|_2 \leq r\}$



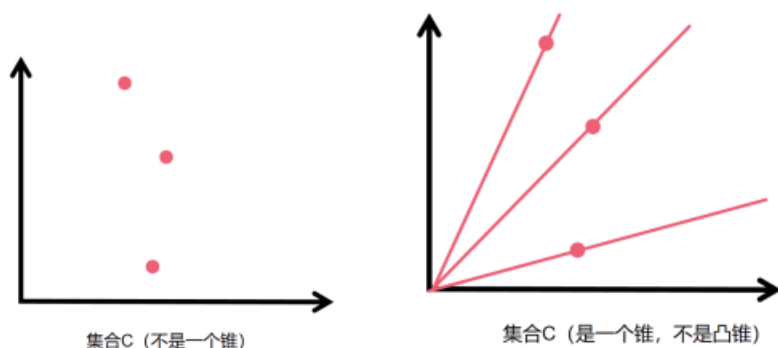
空集也是一个凸集, 一般的点和线也是一个凸集、超平面是一个凸集、超平面的一层所定义的半平面, 也是一个凸集。多边形和球也是一个凸集。

数学基础补充 (非课内)

为了更好的看懂具体的文献中的定义, 需要补充一些概念。摘录自凸优化教材。

## 锥

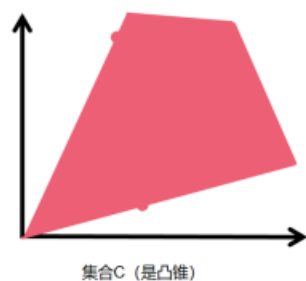
如果对于任意  $x \in C$  和  $\theta \geq 0$  都有  $\theta x \in C$ , 我们称集合  $C$  是锥或者非负齐次。这个理解就是集合中的每个点都有原点引出的一条射线穿过。



我们可以利用凸集的定义和锥的定义推导凸锥的定义，即：

$$\begin{cases} \theta x_1 + (1-\theta)x_2 \in C, \theta \in [0,1] \\ \theta x \in C, \theta \geq 0 \end{cases}$$

由第二式，第一式的参数可以任意替换为  $\theta \geq 0$ ，故有凸集的定义：如果集合  $C$  是锥，并且是凸的，则称  $C$  为凸锥，即对于任意的  $x_1, x_2 \in C$  和  $\theta_1, \theta_2 \geq 0$ ，都有  $\theta_1 x_1 + \theta_2 x_2 \in C$ 。



锥组合的定义略。

集合  $C$  的锥包是  $C$  中所有元素的所有锥组合的集合，即：

$$\{\theta_1 x_1 + \dots + \theta_k x_k \mid x_i \in C, \theta_i \geq 0, i = 1, \dots, k\}$$

它是包含  $C$  的最小的凸锥（如下图所示）

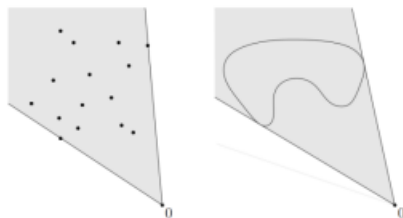


Figure 2.5 The conic hulls (shown shaded) of the two sets of figure 2.3.



### 2.2.3 范数球和范数锥

设  $\|\cdot\|$  是  $\mathbf{R}^n$  中的范数 (参见附录 A.1.2)。由范数的一般性质可知, 以  $r$  为半径,  $x_c$  为球心的范数球  $\{x \mid \|x - x_c\| \leq r\}$  是凸的。关于范数  $\|\cdot\|$  的范数锥是集合

$$C = \{(x, t) \mid \|x\| \leq t\} \subseteq \mathbf{R}^{n+1}.$$

顾名思义, 它是一个凸锥。

**例 2.3** 二阶锥是由 Euclid 范数定义的范数锥, 即

$$\begin{aligned} C &= \{(x, t) \in \mathbf{R}^{n+1} \mid \|x\|_2 \leq t\} \\ &= \left\{ \begin{bmatrix} x \\ t \end{bmatrix} \mid \begin{bmatrix} x \\ t \end{bmatrix}^T \begin{bmatrix} I & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} \leq 0, t \geq 0 \right\}. \end{aligned}$$

二阶锥的其他名字也常常被使用。它由二次不等式定义, 因此也被称为二次锥。同时, 也称其为 Lorentz 锥或冰激凌锥。图 2.10 显示了  $\mathbf{R}^3$  上一个的二阶锥。

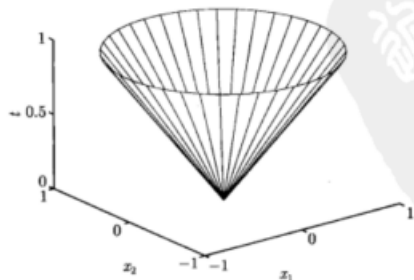


图 2.10  $\mathbf{R}^3$  中二阶锥  $\{(x_1, x_2, t) \mid (x_1^2 + x_2^2)^{1/2} \leq t\}$  的边界

## 多面体

多面体是有限个半空间和超平面的交集。半空间可以看做不等式约束  $a_j^T x \leq b_j$ , 超平面可以看做等式约束  $c_j^T x = d_j$ , 故  $P = \{x \mid a_j^T x \leq b_j, j=1, \dots, m, c_j^T x = d_j, j=1, \dots, p\}$ 。

下图显示了一个由五个半空间的交集定义的多面体。

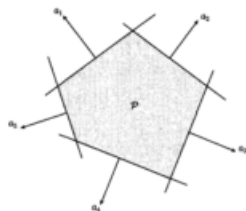


图 2.11 多面体  $P$  (阴影所示) 是外法向量为  $a_1, \dots, a_5$  的五个半空间的交集。

## 半正定锥

我们用  $S^n$  表示实对称  $n$  阶方阵的集合, 即  $S^n = \{X \in \mathbb{R}^{n \times n} \mid X = X^T\}$ 。这是一个维数为  $\frac{n(n+1)}{2}$  的向量空间。我们用  $S_+^n$  表示对称半正定矩阵的集合:

$$S_+^n = \{X \in S^n \mid X \succeq 0\},$$

用  $S_{++}^n$  表示对称正定矩阵的集合:  $S_{++}^n = \{X \in S^n \mid X \succ 0\}$

集合  $S_+^n$  是一个凸锥: 如果  $\theta_1, \theta_2 \geq 0$  并且  $A, B \in S_+^n$ , 那么

例:  $S^2$  上的半正定锥。我们有  $X = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in S_+^2 \Leftrightarrow x \geq 0, z \geq 0, xz \geq y^2$ , 下图显示了这个锥的边界:

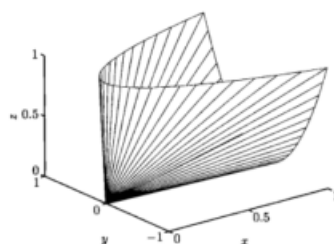


图2.12  $S^2$  中半正定锥的边界。

## 保持凸性的操作

- **Intersection:** the intersection of convex sets is convex.



- **Affine Maps:** if  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  is an affine function (i.e.,  $f(x) = Ax+b$ ) and  $C$  is convex, then  $f(C) = \{Ax+b : x \in C\}$  is convex.

**Example:** an ellipsoid is image of a unit ball after an affine map



- **Perspective Function:** Let  $P: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  be  $P(x,t) = x/t$ , and  $S \in \mathbb{R}^{n+1}$  is convex, then  $P(S)$  is convex.

The perspective function scales or normalizes vectors so the last component is one, and then drops the last component.



1. 两个凸集的交集
2. 对一个凸集进行仿射（线性映射）
3. 透视函数，对向量的某一维进行一个放缩，可以得到一个影像，这个影响还是一个凸集。

## 广义不等式

如果锥  $K \subseteq \mathbb{R}^n$  满足  $K$  是凸的、闭的、实的（具有非空内部）、尖的（不包含直线，或者说  $x \in K, -x \in K \Rightarrow x=0$ ）

正常锥  $K$  可以用来定义广义不等式，即  $\mathbb{R}^n$  上的偏序关系。这种偏序关系和  $\mathbb{R}$  上的标准序有很多相同的性质。用正常锥  $K$  可以定义  $\mathbb{R}^n$  上的偏序关系如下：

$$x \preceq_K y \iff y - x \in K.$$

类似地，我们定义相应的严格偏序关系为  $x \prec_K y \iff y - x \in \text{int } K$ ，我们称其为严格的广义不等式。其中  $\text{int } K$  是指集合  $K$  的内部。

当  $K = \mathbb{R}_+$  时，偏序关系  $\preceq_K$  就是通常意义下  $\mathbb{R}$  中序  $\leq$ ，相应地，严格偏序关系  $\prec_K$  与  $\mathbb{R}$  上的严格序  $<$  相同。因此，广义不等式包含了  $\mathbb{R}$  上的（严格和不严格）不等式，它是广义不等式的一种特殊情况。

朴素的理解： $n$  维空间中，给定一个正常锥  $K$  和需要比较的两个点  $x$  和  $y$ 。 $y-x$  构成一个新的点，这个新的点如果属于在  $K$  内部，则有  $y-x \in \text{int } K \iff x \prec_K y$ ；如果这个点在  $K$  集合中，则  $y-x \in K \iff x \preceq_K y$

半正定锥和矩阵不等式，半正定锥是  $S^n$  空间中的正常锥，相应的广义不等式  $\preceq_K$  就是通常的矩阵不等式，即  $X \preceq_K Y$  等价于  $Y-X$  为半正定矩阵。在矩阵空间  $S^n$  中， $S_+^n$  的内部由正定矩阵组成，因此严格广义不等式也等同于通常的对称矩阵的严格不等式，即  $X \prec_K Y$  等价于  $Y-X$  为正定矩阵。因为我们经常使用它，所以经常省略  $K$  这个下标。

$P$  在对称矩阵空间  $S^n$  中正常锥  $S_+^n$  的内部  $\iff P \succ_K 0 \iff P$  是正定矩阵  $\iff x^T P x > 0$

**例 2.16**  $[0, 1]$  上非负的多项式锥。  $K$  定义如下

$$K = \{c \in \mathbb{R}^n \mid c_1 + c_2 t + \cdots + c_n t^{n-1} \geq 0 \text{ 对于 } t \in [0, 1]\}, \quad (2.15)$$

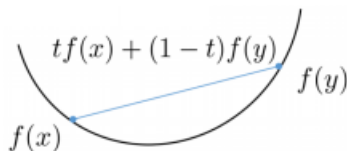
即  $K$  是  $[0, 1]$  上最高  $n-1$  阶的非负多项式（系数）锥。可以看出  $K$  是一个正常锥，其内部是  $[0, 1]$  上为正的多项式的系数集合。

两个向量  $c, d \in \mathbb{R}^n$  满足  $c \preceq_K d$  的充要条件是，对于所有  $t \in [0, 1]$  有

$$c_1 + c_2 t + \cdots + c_n t^{n-1} \leq d_1 + d_2 t + \cdots + d_n t^{n-1}.$$

## 凸函数

一个函数  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  是凸的，当且仅当对于任意  $t \in [0, 1]$ ，有  $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$ 。即任意两点的连线，在函数所包括的范围之内。



凸函数的例子：

- Exponential function:  $e^{ax}$
- logarithmic function  $\log(x)$  is concave
- Affine function:  $a^T x + b$
- Quadratic function:  $x^T Q x + b^T x + c$  is convex if  $Q$  is positive semidefinite (PSD)
- Least squares loss:  $\|y - Ax\|_2^2$
- Norm:  $\|x\|$  is convex for any norm

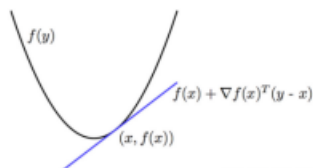
$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \quad \|x\|_1 = \sum_{i=1}^n |x_i|$$

### First Order Convexity Conditions

#### Theorem

Suppose  $f$  is differentiable. Then  $f$  is convex if and only if for all  $x, y \in \text{dom } f$

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$



使用一阶泰勒展开，在某个点上找函数的切线，切线是不可能穿过函数的，一定是在凸函数的一侧。

### Second Order Convexity Conditions

Suppose  $f$  is twice differentiable. Then  $f$  is convex if and only if for all  $x \in \text{dom } f$

$$\nabla^2 f(x) \succeq 0$$

二阶导数判断凸性：二阶导的矩阵是半正定的。

怎么证明局部最优点就是全局最优点呢？如果我们找到一个局部最优点，其导数是 0。我们按照泰勒展开式，根据刚才的一阶判定条件。在导数等于 0 的时候， $x$  一定是最优值的。如

果导数不等于 0，我们可以找到一个使得导数等于 0 的方向移动，直到导数等于 0。

## 常见的凸优化问题

### 1. 线性规划

可行域是多边形（凸集），目标函数是一个线性的仿射函数，所以是一个凸优化问题。  
线性规划的形式：

$$\begin{aligned} &\text{minimize} && c^T x + d \\ &\text{subject to} && Gx \preceq h \\ &&& Ax = b \end{aligned}$$

### 2. 几何优化问题

单项式(monomial)：每一位做一个指数相乘，再乘以一个系数，换句话说具有以下形式：

$$f(x) = cx_1^{a_1} x_2^{a_2} \dots x_n^{a_n}, \text{ where } c \geq 0, a_i \in \mathbb{R}$$

多项式(posynomial)：一系列单项式的和。

形式：

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 1, \quad i = 1, \dots, m \\ &&& h_i(x) = 1, \quad i = 1, \dots, p \end{aligned}$$

其中  $f_i$  是多项式， $h_i$  是单项式。

几何规划的特征：目标函数和所有的不等式约束都是正多项式，等式约束是一个单项式。  
(几何规划问题是围成的体积的问题。)

Eg: 生产手提箱

决定一个手提箱形状的有长宽高三个变量，我们优化的目标是希望花费的材料越少越好，材料的总数是和表面积正相关的，所以我们希望最小化表面积。

A manufacturer is designing a suitcase (手提箱)

- Variables:  $h, w, d$
- Want to **minimize surface area**:  $2(hw + hd + wd)$  (i.e. amount of material used)
- Have a **target volume**:  $hwd \geq 5$
- Practical/aesthetic (美学) constraints limit aspect ratio:  
 $h/w \leq 2, h/d \leq 3$
- Constrained by airline to  $h + w + d \leq 7$

$\begin{aligned} &\text{minimize} && 2hw + 2hd + 2wd \\ &\text{subject to} && h^{-1}w^{-1}d^{-1} \leq \frac{1}{5} \\ &&& hw^{-1} \leq 2 \\ &&& hd^{-1} \leq 3 \\ &&& h + w + d \leq 7 \\ &&& h, w, d \geq 0 \end{aligned}$	$\begin{aligned} \tilde{h} &= \log h \\ \tilde{w} &= \log w \\ \tilde{d} &= \log d \end{aligned}$ $\Rightarrow$	$\begin{aligned} &\text{minimize} && 2e^{\tilde{h}+\tilde{w}} + 2e^{\tilde{h}+\tilde{d}} + 2e^{\tilde{w}+\tilde{d}} \\ &\text{subject to} && e^{-\tilde{h}-\tilde{w}-\tilde{d}} \leq \frac{1}{5} \\ &&& e^{\tilde{h}-\tilde{w}} \leq 2 \\ &&& e^{\tilde{h}-\tilde{d}} \leq 3 \\ &&& e^{\tilde{h}} + e^{\tilde{w}} + e^{\tilde{d}} \leq 7 \end{aligned}$
--	--	---

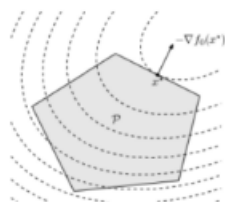
我们把除法全部换成乘法，把大于号全部换成小于号。去掉非负的条件，就用指数进行代换。

### 3. 二次规划

定义：目标函数是凸二次型并且约束函数为仿射，此问题为二次规划问题（QP 问题）。其形式如下：

$$\begin{aligned} & \text{minimize} && (1/2)x^T Px + q^T x + r \\ & \text{subject to} && Gx \preceq h \\ & && Ax = b \end{aligned}$$

其中  $P \in S_+^n$ （ $P$  是  $n$  阶对称半正定矩阵）， $G \in R^{m \times n}$ ， $A \in R^{p \times n}$ ， $x \in R^{n \times 1}$ ，图像如下：




例子：最小二乘

**Constrained Least Squares**

Given a set of measurements  $(x_1, y_1), \dots, (x_m, y_m)$ , where  $x_i \in R^n$  is the  $i$ -th input and  $y_i \in R$  is the  $i$ -th output, fit a linear function minimizing mean square error, subject to known bounds on the linear coefficients.

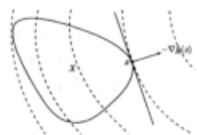
minimize  $\|Ax - b\|_2^2 = x^T A^T A x - 2b^T A x + b^T b$   
subject to  $l_i \leq x_i \leq u_i$  for  $i = 1, \dots, n$ .



如果不等式约束也是凸二次型，即：

$$\begin{aligned} & \text{minimize} && (1/2)x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} && (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \\ & && Ax = b \end{aligned}$$

其中  $P_i \in S_+^n$ ， $i = 0, 1, \dots, m$ ，这一问题称为二次约束二次规划（QCQP）问题，在 QCQP 中，如果  $P_i \succ 0$ ，我们在椭圆的交集构成的可行集上极小化凸二次函数，如下图所示。



线性规划是二次规划的特例，二次规划是二次约束二次规划的特例。

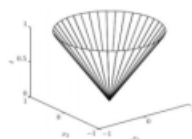
## 4. 锥优化

线性目标函数，可行域变成了锥形的。即有如下形式：

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax + b \in K \end{aligned}$$

其中  $K$  是一个凸锥，（比如：  $n$  维正向量（即  $n \times 1$  矩阵），半正定矩阵）。

举例来说，二阶锥优化中， $K$  是一个二阶锥，即  $K = \{(x, t) \mid \|x\|_2 \leq t\}$



$t$  变大， $x$  的范围也变大，构成了一个锥形。关于  $x$  的线性组合（目标函数）在一个锥形里，这个就叫锥优化。

## 广义不等式约束

凸优化问题的形式如下：

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && a_i^T x = b_i, \quad i = 1, \dots, p \end{aligned}$$

其中  $f_0, \dots, f_m$  为凸函数。等式约束函数  $h_i(x) = a_i^T x - b_i$  是仿射的。

通过将不等式约束函数扩展为向量并使用广义不等式，可以得到标准形式凸优化问题（上式）的一个非常有用的推广。

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \preceq_{K_i} 0, \quad i = 1, \dots, m \\ & && Ax = b \end{aligned}$$

其中  $f_0: \mathbb{R}^n \rightarrow \mathbb{R}$ ， $K_i \subseteq \mathbb{R}^{k_i}$  为正常锥， $f_i: \mathbb{R}^n \rightarrow \mathbb{R}^{k_i}$  为  $K_i$ -凸的。此问题为标准形

式的广义不等式意义下的凸优化问题。第一式是当  $K_i = \mathbb{R}_+$  时的特殊情况。

在广义不等式的凸优化问题中，最简单的是锥规划问题，它有线性目标函数和一个不等式约束函数，该函数是仿射的（因此是  $K$ -凸的），形式如下：

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Fx + g \preceq_K 0 \\ & && Ax = b \end{aligned}$$

当  $K$  为非负象限时，锥形式问题退化为线性规划。我们可以将锥形式问题视为线性规划的推广，其中的分量不等式被替换为广义线性不等式。

## 4. 半正定规划 (Semi-Definite Programming, SDP)

当  $K$  为  $S_+^k$  ( $K$  是  $k$  阶对称半正定矩阵)，即  $K$  为  $k$  阶半正定矩阵锥，相应的锥形式问题称为半正定规划 (SDP)。形式为：

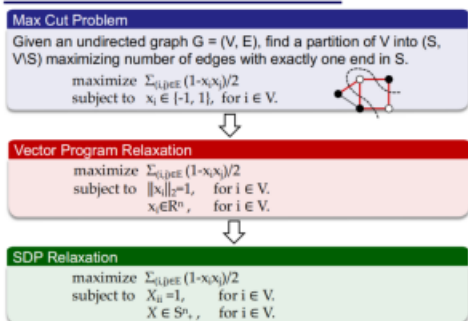
$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && x_1 F_1 + \cdots + x_n F_n + G \preceq 0 \\ & && Ax = b \end{aligned}$$

其中  $G, F_1, \dots, F_n \in S^k, A \in R^{p \times n}$ 。这里的不等式是线性矩阵不等式 (LMI)。

$A(x) = x_1 A_1 + \cdots + x_n A_n \preceq B$  称为关于  $x$  的线性矩阵不等式 (LMI)，其中  $B, A_i \in S^m$  ( $m$  阶对称矩阵)。线性矩阵不等式的解  $\{x \mid A(x) \preceq B\}$  是凸集，它是半正定锥在由  $f(x) = B - A(x)$  给定的仿射映射  $f: R^n \rightarrow S^m$  下的原象。换句话说， $B - A(x)$  定义了一个  $S^m$  中的一个点，如果  $B - A(x)$  在半正定锥中（包含表面），那么就接受  $n$  维空间中的原点  $x$ 。

如果矩阵  $G, F_1, \dots, F_n$  都是对角阵，那么上式的 LMI 等价于  $n$  个线性不等式，SDP 退化为线性规划。

### Example: SDP



对于上例这个最大切问题，我们希望能够找到给定一个图的最大切，也就是找到一个划分，使得连接黑白结点（两个划分）的边数最大。我们令黑色结点为 -1，令白色结点为 1，



这样的话可以使得那些连接两个划分的边权为  $x_i x_j = -1$ ，即真正对  $\sum_{(i,j) \in E} (1 - x_i x_j) / 2$  这个目标函数产生贡献。

但是整数约束的规划是一个比较复杂的问题，难以求解。我们尝试做一个放缩，找到一个近似的原问题。我们把  $x_i$  看成一个向量，这个向量的长度是 1，这样放缩可能找到原问题

的近似解或者原始解。我们构造  $x_i$  的点积生成的矩阵，即  $X = \begin{bmatrix} \vec{x}_1 \cdot \vec{x}_1 & \cdots & \vec{x}_1 \cdot \vec{x}_n \\ \vdots & \ddots & \vdots \\ \vec{x}_n \cdot \vec{x}_1 & \cdots & \vec{x}_n \cdot \vec{x}_n \end{bmatrix}$ ，我们

发现其对角线元素  $X_{ii} = 1$ ，至少满足放缩前的条件  $\|\vec{x}_i\|_2 = 1$ 。

<https://www.cs.cmu.edu/~anupamg/adv-approx/lecture14.pdf>

## 凸优化的对偶性

表示一个凸集有两种等价的方法：

1. 凸集中点的集合（标准表示方法）
2. 包含这个凸集的半空间的集合（对偶表示方法）

一个封闭的凸集  $S$  是所有封闭的半空间的交集。