计算方法

刘景铖

计算机软件新技术国家重点实验室
南京大学

回顾

上节课:

- 随机游走的碰撞时间(**Hitting time**)和返程时间(**commute time**)
- 线性规划 **LP**，顶点的不同定义

这节课:

- 对偶性
- 零和游戏

LP的顶点

考虑$\max\langle c, x\rangle$，约束在$P := \{Ax \leq b\}$这个polytope内

顶点可以有3种定义：

1. **边角(corner)**: 如果不存在$y \neq 0$ 使得 $x + y \in P$ and $x - y \in P$，则称点x是一个边角
2. **极值点:** 如果 $\exists c$ 使得x是该目标方向c的唯一最优解，则称点x是一个极值点.
3. **基本解**: 如果 $\langle A_i, x\rangle = b_i$ ，我们称第i个约束是紧致 (tight)的，其中 A_i 是第i行.

   对于给定的$x \in P$, 记 $A^=$ 为A 中关于 x 紧致的约束组成的子矩阵.

   如果$A^=$ 是满秩的, i.e. $rank(A^=) = n$ ，那么我们称x是一个基本解.

事实上，这3种定义是等价的。

LP的稳定性：对c的扰动，对A的扰动

顶点的个数：一般情况下面可能是指数级别的, $\binom{m}{n}$

LP的顶点

考虑$\max\langle c,x\rangle$，约束在$P := \{Ax \le b\}$这个polytope内

1.**边角(corner)**: 如果不存在$y \neq 0$ 使得 $x+y \in P$ and $x-y \in P$，则称点x是一个边角

3.**基本解**: 紧致的约束组成的子矩阵$A^=$ 是满秩的, i.e. $rank(A^=) = n$

1) $\Rightarrow$ 3): 或者说是$\neg$3) $\Rightarrow$ $\neg$1)

假设存在 $rank(A^=) < n, i.e.,\ \exists y \neq 0, A^= y = 0$

考虑$x+\epsilon y, x-\epsilon y$，注意到

$$A^=(x+\epsilon y) = A^= x$$

$$A^=(x-\epsilon y) = A^= x$$

取ϵ足够小，使得其它的不等式约束不被违反

则有$x+\epsilon y \in P, x-\epsilon y \in P$

LP的顶点

考虑$\max\langle c, x\rangle$，约束在$P \coloneqq \{Ax \leq b\}$这个polytope内

1.**<u>边角(corner)</u>**: 如果不存在$y \neq 0$ 使得 $x + y \in P$ and $x - y \in P$，则称点x是一个边角

3.**<u>基本解</u>**: 紧致的约束组成的子矩阵$A^{=}$ 是满秩的, i.e. $rank(A^{=}) = n$

3) $\Rightarrow$ 1): 或者说是$\neg$1) $\Rightarrow$ $\neg$3)

假设有$y \neq 0$使得$x + y \in P, x - y \in P$

$$A^{=}(x + y) \leq b^{=}$$

$$A^{=}(x - y) \leq b^{=}$$

其中 $A^{=}x = b^{=}$

因此有 $A^{=}y \leq 0, A^{=}y \geq 0$进而 $A^{=}y = 0$

LP的顶点

考虑$\max\langle c, x\rangle$，约束在$P := \{Ax \le b\}$这个polytope内

顶点可以有3种等价的定义：

1. **<u>边角(corner)</u>**: 如果不存在$y \neq 0$ 使得 $x + y \in P$ and $x - y \in P$，则称点x是一个边角

2. **<u>极值点:</u>** 如果 $\exists c$ 使得x是该目标方向c的唯一最优解，则称点x是一个极值点.

3. **<u>基本解</u>**: 如果 $\langle A_i, x\rangle = b_i$ ，我们称第i个约束是<u>紧致 (tight)</u>的，其中 A_i 是第i行.

   对于给定的$x \in P$, 记 $A^{=}$ 为A 中关于 x 紧致的约束组成的子矩阵.

   如果$A^{=}$ 是满秩的, i.e. $rank(A^{=}) = n$， 那么我们称x是一个基本解.

Simplex算法：从一个顶点开始；找下一个顶点，如果目标函数更优，则选择移到该顶点；重复；

邻居的选择: 最多(m-n)n

如果所有邻居都更差，则当前必定是最优的解：对于凸优化问题，局部最优即是全局最优

最坏情况下面， Simplex算法可能需要指数时间。但是实践中表现往往不错，smoothed analysis

多项式算法： Ellipsoid algorithm，interior point methods

Perfect bipartite matching

定理：考虑二分图完美匹配的线性规划：

$\max \sum_{e\in E} c_e x_e$ subject to

$$\sum_{e\in\delta(v)} x_e = 1 \quad \forall v \in V$$

$$0 \le x_e \le 1 \quad \forall e \in E$$

该LP的最优解一定是整数解

注意：这个LP在一般图上面并不一定是整数解的

Ellipsoid algorithm, separation oracle (选讲)

最小生成树的一种LP写法

- $\max \sum_{e\in E} c_e x_e$
- $\sum_{e\in E(S)} x_e \leq |S| - 1, \quad \forall S \subset V$
- $\sum_{e\in E(V)} x_e = |V| - 1$
- $0 \leq x_e \leq 1$

虽然是指数大小的LP，但是Ellipsoid algorithm只需要有separation oracle，也能多项式时间解出来

Duality：给目标函数证明上界

$$\max 5x_1 + 4x_2$$

$$2x_1 + x_2 \leq 100$$

$$x_1 \leq 30$$

$$x_2 \leq 60$$

$$x_1, x_2 \geq 0.$$

- 如何证明目标函数的一个下界？

Duality：给目标函数证明上界

$$\max 5x_1 + 4x_2$$

$$2x_1 + x_2 \le 100 \quad (1)$$

$$x_1 \le 30 \qquad (2)$$

$$x_2 \le 60 \qquad (3)$$

$$x_1, x_2 \ge 0.$$

如何证明目标函数的一个上界?

- 考虑 $0\times(1) + 5\times(2) + 4\times(3)$
- 考虑 $3\times(1) + 0\times(2) + 1\times(3)$
- 考虑 $\frac{5}{2}\times(1) + 0\times(2) + \frac{3}{2}\times(3)$
- …
- 更一般地，考虑$y_1\times(1) + y_2\times(2) + y_3\times(3)$

Duality：给目标函数证明上界

$$\min 100y_1 + 30y_2 + 60y_3$$
$$2y_1 + y_2 \geq 5$$
$$y_1 + y_3 \geq 4$$
$$y_1, y_2, y_3 \geq 0.$$

$$5x_1 + 4x_2 \leq (2y_1 + y_2)x_1 + (y_1 + y_3)x_2 \leq 100y_1 + 30y_2 + 60y_3$$

- 拉格朗日乘数法
- 弱对偶性 (weak duality)

Primal-Dual

原始(Primal)问题：

$$\max_x \begin{pmatrix} \cdots & c^T & \cdots \end{pmatrix} \begin{pmatrix} \vdots \\ x \\ \vdots \end{pmatrix}$$

$$\text{subject to} \begin{pmatrix} & \vdots & \\ \cdots & A & \cdots \\ & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ x \\ \vdots \end{pmatrix} \leq \begin{pmatrix} \vdots \\ b \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ x \\ \vdots \end{pmatrix} \geq 0$$

对偶(Dual)问题：

$$\min_y \begin{pmatrix} \cdots & b^T & \cdots \end{pmatrix} \begin{pmatrix} \vdots \\ y \\ \vdots \end{pmatrix}$$

$$\text{subject to} \begin{pmatrix} & \vdots & \\ \cdots & A^T & \cdots \\ & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ y \\ \vdots \end{pmatrix} \geq \begin{pmatrix} \vdots \\ c \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ y \\ \vdots \end{pmatrix} \geq 0$$

Weak duality

$$\max\ \langle c, x\rangle$$
$$Ax \le b$$
$$x \ge 0.$$

(Primal LP)

$$\max\ \langle c, x\rangle$$
$$\langle a_1, x\rangle \le b_1$$
$$\dots$$
$$\langle a_m, x\rangle \le b_m$$
$$x \ge 0.$$

$$\min\ \langle b, y\rangle$$
$$\sum_{i=1}^{m} y_i a_i \ge c$$
$$y \ge 0.$$

$$\min\ \langle b, y\rangle$$
$$A^T y \ge c$$
$$y \ge 0.$$

(Dual LP)

弱对偶性 (weak duality)定理: 对于任意原始的最大化LP中的可行解x，与对偶的最小化LP中的可行解y, 都有

$$\langle c, x\rangle \le \langle b, y\rangle$$

即，原始最大值≤对偶最小值.

证明：

$$\langle c, x\rangle = c^T x \le (y^T A)x = y^T(Ax) \le y^T b = \langle b, y\rangle.$$

Complementary Slackness Conditions

如何证明x是最优解？

- 弱对偶定理告诉我们，一个证明方法是：找到对偶LP中的y使得$\langle c, x\rangle = \langle b, y\rangle$

这样的y是否总是存在？

- 强对偶性定理 (strong duality)

什么时候可以取到等号？

- $c^T x \le (y^T A)x$要取到等号：如果$x_j > 0$, 则$\langle a^j, y\rangle = c_j$, 其中 a^j 是A的第j列.
- $y^T(Ax) \le y^T b$要取到等号：如果$y_i > 0$, 则$\langle a_i, x\rangle = b_i$, 其中 a_i 是A的第i行

匈牙利算法

Min-max定理例子：组合优化

二分图中的最大匹配和最小顶点覆盖 (Kőnig's theorem)

$$\max \sum_{e \in E} x_e$$

$$x(\delta(v)) \leq 1 \qquad \forall v \in V$$

$$x_e \geq 0 \qquad \forall e \in E$$

$$\min \sum_{v \in V} y_v$$

$$y_u + y_v \geq 1 \qquad \forall e \in E$$

$$x_v \geq 0 \qquad \forall v \in V$$

整数解分别对应最大匹配和最小顶点覆盖

练习： 在二分图中，可以证明它们的最优解是整数解(integral)

拓展：试推导组合中的Hall’s theorem

因此强对偶定理告诉我们， 二分图中的最大匹配的大小=最小顶点覆盖的大小

Min-max定理例子：最大流最小割

$$\max f_{ts}$$

$$f\left(\delta^{in}(v)\right) - f\left(\delta^{out}(v)\right) \le 0 \qquad \forall v \in V$$

$$f_e \le 1 \qquad \forall e \in E$$

$$f_e \ge 0 \qquad \forall e \in E$$

$$\min \sum_{e \in E} d_e$$

$$d_{uv} + y_u - y_v \ge 0 \qquad \forall uv \in E$$

$$y_s - y_t \ge 1$$

$$y_v \ge 0 \qquad \forall v \in V$$

- **原始问题：** 最大流
- 对偶问题：最小割
- 可证明最优解是整数解

零和游戏

一个矩阵A
- “行玩家”选一行r
- “列玩家”选一列c

行玩家得到的回报是$A_{r,c}$，列玩家得到$-A_{r,c}$

	P	S	R
P	0	-1	1
S	1	0	-1
R	-1	1	0

行玩家的目标是最大化 $A_{r,c}$，而列玩家目标是最小化$A_{r,c}$

纳什均衡：即使一个玩家知道对方的策略之后，他/她也不能找到比当前策略严格更优的策略。

单纯的策略(pure strategy)：选一行/列
混合的策略(mixed strategy)：单纯策略的一个概率分布

Von-Neumann Minimax Theorem

给定$A \in \mathbb{R}^{m\times n}$
假设行玩家策略的概率分布是 $x \in \Delta^m$
列玩家策略的概率分布是$y \in \Delta^n$

那么它们玩下来的期望回报是$x^T Ay$

Von-Neumann Minimax Theorem.

$$\max_{x\in\Delta^m} \min_{y\in\Delta^n} x^T Ay = \min_{y\in\Delta^n} \max_{x\in\Delta^m} x^T Ay$$

不管谁先宣布自己的策略，都会达到均衡解

Von-Neumann Minimax Theorem

Von-Neumann Minimax Theorem.

$$\max_{x\in\Delta^m}\min_{y\in\Delta^n} x^T A y = \min_{y\in\Delta^n}\max_{x\in\Delta^m} x^T A y$$

证明:

左边: 行玩家先选定 x

列玩家在给定$x^T A$后，如何选择最优的y?

$x^T A$为一行向量， 注意到y为概率分布

令 $(x^T A)^{(j)}$为$x^T A$的第 j列，则有

$$\min_j (x^T A)^{(j)} \le x^T A y \le \max_j (x^T A)^{(j)}$$

且等号可以取到。因此$\min_{y\in\Delta^n} x^T A y = \min_j (x^T A)^{(j)}$

左边即可化简为$\max_{x\in\Delta^m}\min_j (x^T A)^{(j)}$

Von-Neumann Minimax Theorem

<u>Von-Neumann Minimax Theorem</u>.

$$\max_{x\in\Delta^m}\min_{y\in\Delta^n} x^T A y = \min_{y\in\Delta^n}\max_{x\in\Delta^m} x^T A y$$

证明(cont'd)：

左边可化简为$\max_{x\in\Delta^m}\min_j (x^T A)^{(j)}$，这可以通过LP表示

引入辅助变量 $t = \min_j (x^T A)^{(j)}$

$$\max \quad t$$

$$(x^T A)^{(j)} = \sum_{i=1}^m x_i a_{ij} \geq t \qquad \forall j = 1, \dots, n$$

$x \in \Delta^m$

$$\sum_{i=1}^m x_i = 1$$

$$x_i \geq 0 \qquad \forall i = 1, \dots, m$$

Von-Neumann Minimax Theorem

Von-Neumann Minimax Theorem.

$$\max_{x\in\Delta^m}\min_{y\in\Delta^n} x^T A y = \min_{y\in\Delta^n}\max_{x\in\Delta^m} x^T A y$$

证明(cont'd)：

类似地，右边可化简为$\min\limits_{y\in\Delta^n}\max\limits_i (Ay)_i$，进而通过LP表示

引入辅助变量 $r = \max\limits_i (Ay)_i$

$$\min \quad r$$

$$(Ay)_i = \sum_{j=1}^{n} a_{ij}\, y_j \le r \qquad \forall i = 1, \dots, m$$

$y \in \Delta^n$

$$\sum_{i=1}^{n} y_i = 1$$

$$y_i \ge 0 \qquad \forall i = 1, \dots, n$$

Yao’s Minimax Principle

一个随机算法的最坏运行时间，是它在所有输入实例中期望运行时间的最大值

试考虑这样的一个零和游戏：

“算法”玩家需要从不同的算法中选择，目标是最小化运行时间

“攻击”玩家需要从不同的输入实例中选择，目标是最大化运行时间

“**算法**”的**混合策略**：随机算法

“攻击”的混合策略：输入实例的概率分布

要成功地“攻击”所有的随机算法，等价的问题是：

给出一个输入实例的概率分布，使得所有的确定性算法的期望运行时间都比较大