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

广告：暑期讲习班

- https://tcs.nju.edu.cn/wiki/index.php?title=%E8%AE%A1%E7%AE%97%E7%90%86%E8%AE%BA%E4%B9%8B%E7%BE%8E_(Summer_2024)
- 第三届“计算理论之美”暑期讲习班
- 2024年7月13日至7月16日
- 主题：算法设计

回顾

上节课:

- 电路电阻网络
- “等效电阻”距离

这节课:

- 随机游走的碰撞时间(**Hitting time**)和返程时间(**commute time**)
- 线性规划 **LP**

回顾：随机游走

1. <u>碰撞时间 (Hitting time):</u> $H_{u,v} := \min\{t \geq 1 \mid X_1 = u \ and \ X_t = v\}$ and $h_{u,v} = \mathbb{E}[H_{u,v}]$.

2. <u>返程时间 (Commute time)</u>: $C_{u,v} := h_{u,v} + h_{v,u}$.

3. <u>遍历时间 (Cover time):</u> cover_v 定义为：从v出发的随机游走访问每个节点至少一次需要的期望时间；$\text{cover}_G := \max_{v} \text{cover}_v$

Commute time

定理. 对任意的节点s 和 t，$C_{s,t} = 2mR_{\mathrm{eff}}(s,t)$，其中$m = |E(G)|$.
证明：固定节点 t ，记$h_{u,t}$为从点 u到节点 t的hitting time，满足$\forall u \neq t$

$$h_{u,t} = 1 + \frac{1}{d_u}\sum_{v\sim u} h_{v,t} \Rightarrow d_u h_{u,t} - \sum_{v\sim u} h_{v,t} = d_u$$

考虑$\overrightarrow{h_{*,t}}$这一向量，它满足

$$\begin{pmatrix} & D-A & \\ & & \end{pmatrix}\begin{pmatrix} h_{u,t} \\ h_{t,t} \end{pmatrix} = \begin{pmatrix} d_u \\ d_t - 2m \end{pmatrix}$$

(To be cont'd..)

Commute time

<u>定理</u>. 对任意的节点s 和 t，$C_{s,t} = 2mR_{\text{eff}}(s,t)$，其中$m = |E(G)|$.
证明：固定节点 s，记$h_{u,s}$为从点 u到节点 s的hitting time，满足$\forall u \neq s$

$$h_{u,s} = 1 + \frac{1}{d_u}\sum_{v\sim u} h_{v,s} \Rightarrow d_u h_{u,s} - \sum_{v\sim u} h_{v,s} = d_u$$

考虑$\overrightarrow{h_{*,s}}$这一向量，它满足

$$\begin{pmatrix} & & \\ & D-A & \\ & & \end{pmatrix}\begin{pmatrix} h_{s,s} \\ h_{u,s} \\ \\ h_{t,s} \end{pmatrix} = \begin{pmatrix} d_s - 2m \\ d_u \\ \\ d_t \end{pmatrix}$$

(To be cont'd..)

定理. 对任意的节点s 和 t，$C_{s,t} = 2mR_{\mathrm{eff}}(s,t)$，其中$m = |E(G)|$.

证明(cont'd)：

$$L(\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}}) = \begin{pmatrix} d_s \\ d_u \\ \vdots \\ d_t - 2m \end{pmatrix} - \begin{pmatrix} d_s - 2m \\ d_u \\ \vdots \\ d_t \end{pmatrix} = \begin{pmatrix} 2m \\ 0 \\ \vdots \\ -2m \end{pmatrix}$$

因此$\frac{L(\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}})}{2m} = b_{s,t}$。回顾$L\phi = b_{st}$，有解，且解空间是一维的

令$\phi = \frac{\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}}}{2m}$，有

$$R_{\mathrm{eff}}(s,t) = \phi(s) - \phi(t) = \frac{h_{s,t} - h_{s,s}}{2m} - \frac{h_{t,t} - h_{t,s}}{2m} = \frac{h_{s,t} + h_{t,s}}{2m} = \frac{C_{s,t}}{2m}$$

Cover time

推论. $C_{u,v} \leq 2m$，对于任意的$uv \in E$成立.

定理. 连通图的遍历时间最多为$2m(n-1)$.

Approximating Cover Time by Resistance Diameter

Theorem. Let $R(G) \coloneqq \max_{u,v} R_{\mathrm{eff}}(u,v)$ be the resistance diameter. Then,

$$m \cdot R(G) \ \leq \ \mathrm{cover}(G) \leq \ 2e^3 m \cdot R(G) \cdot \ln n + n$$

Proof: Firstly,

$$\mathrm{cover}(G) \geq \max\{h_{uv}, h_{vu}\} \geq \frac{C_{uv}}{2} = mR_{uv} \ ,$$

which is the lowerbound.

For the upperbound, notice that the maximum commute time from any vertex is at most $2mR(G)$

If the random walk is run for $2e^3 m \cdot R(G)$, by Markov's inequality, the probability that a vertex is not visited is at most $1/e^3$

If we repeat this $\ln n$ times, the probability that a vertex is not visited is at most $1/n^3$

By a union bound, the probability that there exists a vertex not visited is at most $1/n^2$

In such cases, we can pay for another pessimistic cover time of n^3

Combined, we have $\mathrm{cover}(G) \leq \ 2e^3 m \cdot R(G) \cdot \ln n + \frac{1}{n^2} n^3$

Graph Connectivity (选讲)

Theorem. There is an $O(n^3)$ time algorithm to solve s-t connectivity using only $O(\log n)$ space
Using random walk, the space requirement is $O(\log n)$ and expected running time is $O(|V||E|) = O(n^3)$

You may wonder, is randomness necessary for checking graph connectivity in log-space?
Definition. A sequence σ is (d, n)-universal if for every labeled connected d-regular graphs and every starting vertex s, the walk defined by σ started from s covers every vertices

Theorem. There exists (d, n)-universal sequence of length $O(n^3 d^2 \log nd)$ for undirected graphs
HINT: Cover time is at most $O(n^2 d)$ for d-regular graphs

Reingold's Theorem For undirected graphs, one can explicitly construct such a universal sequence in log-space

It is an open problem to derandomize log-space connectivity
Though likely not through "directed" universal sequences

更多的联系与应用

- Spectral Embedding/Partitioning
  - 课堂上：拉普拉斯矩阵的特征值 与 图的连通性的关系
  - 上次作业：使用拉普拉斯矩阵的特征向量画图, elastic spring networks
  - 更一般地，拉普拉斯矩阵的特征向量包含的信息可用于做图的分解/分割
  - Cheeger’s inequality and its generalizations
- 图的稀疏化 (Graph sparsification)
  - 等效电阻与均匀随机生成树
  - 根据等效电阻，对边进行采样
  - 更复杂的算法可以做到O(n)条边
- 计算等效电阻: 解拉普拉斯方程组
  - 几乎线性时间里面求解;
  - 作为对比：一般的线性方程组，高斯消元需要 $O(n^3)$；迭代法则需要条件数比较好；
- 网络流问题：近似线性时间
- ……

开放研究问题：如果对有向图增加或删掉一条边，Pagerank会变化多少？

Optimization

给定连续可导$f\colon \mathbb{R}^n \to \mathbb{R}$, 最小化$f$

例子：

最小二乘法：$f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$

向量投影：$f(\boldsymbol{c}) = \left\|\boldsymbol{c}\,\vec{\boldsymbol{a}} - \vec{\boldsymbol{b}}\right\|_2^2$

求伪逆(Pseudoinverse)：$f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$, subject to $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$

对称矩阵的特征值：$f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A}\,\boldsymbol{x}$, subject to $x^\top x = 1$

线性规划：$\boldsymbol{f(x) = c^\top x}$, subject to $\boldsymbol{Ax \geq b}$

主成分分析 (Principal component analysis)：$f(\boldsymbol{C}) = \left\|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{C}^\top \boldsymbol{X}\right\|_{\boldsymbol{F}}$, subject to $C\,C^\top = I_{d\times d}$

离散组合优化：最小生成树，最小割、最大流（网络流），最短路径，最大匹配；最大独立集、团、支配集，最小覆盖，背包问题，最大割，max-SAT，旅行商问题，最长路径（哈密顿路径）……

线性规划 (Linear Programming)

例子：面包店

	面粉	糖	鸡蛋
甜甜圈 (Donuts)	2	2	7
蛋糕 (Cake)	5	9	12
现有原料	≤ 200	≤ 300	≤ 500

甜甜圈单个的利润=5， 蛋糕单个的利润=25

在现有的原料的限制下，如何最大化利润？

线性规划 (Linear Programming)

例子： 电力供应

已知：发电厂传输电力到各个城市的电力损耗，和各个城市的电力需求

	城市A	城市B	城市C
发电厂1	4%	5%	1%
发电厂2	3%	2%	7%
电力需求	≥ 40	≥ 60	≥ 80

在满足电力需求的前提下，如何最小化电力损失？

线性规划的矩阵形式

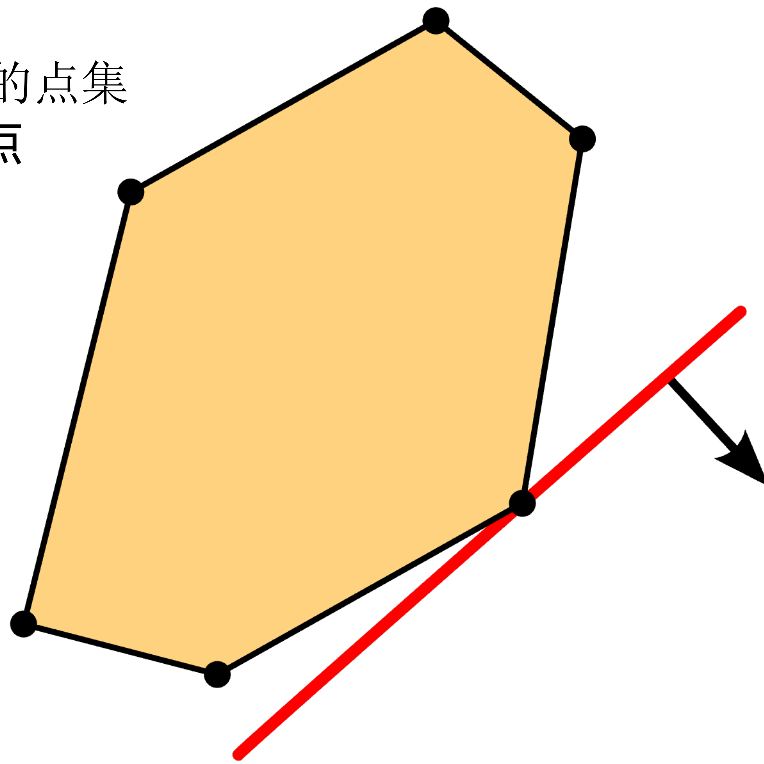
- 线性的约束可以通过矩阵表示： 如果有m个约束和n个变量, 则可以有$A \in \mathbb{R}^{m\times n}$， 它的第$i$行正是第$i$个约束
- 目标函数亦可以通过向量$c \in \mathbb{R}^n$表示

$$\max \langle c, x\rangle$$
$$Ax \le b$$

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

几何直观

Polytope (多胞体): LP的可行区域，即满足所有约束的点集
Vertex/Corner (顶点): 可行区域中n个超平面相交的点



Source: Wikipedia CC0

每个约束定义一个半平面 (halfspace)

目标函数定义了一个方向：
找到该方向上最“远”的顶点（可能不唯一）

事实1: LP的可行区域总是凸的 (convex)
事实2: LP一定存在一个最优解在顶点上

凸集 (convex set): $\forall x, y \in S, \forall t \in [0,1], tx + (1-t)y \in S$

整数线性规划

很多组合优化问题都能通过**整数**线性规划表示出来

最大二分图匹配：

- $\max \sum_{e\in E} c_e x_e$
- $\sum_{e\in\delta(v)} x_e \le 1$ $\quad\forall v \in V$
- $x_e \in \{0,1\}$ $\quad\forall e \in E$

最大独立集：

- $\max \sum_{v\in V} c_v x_v$
- $x_u + x_v \le 1$ $\quad\forall uv \in E$
- $x_v \in \{0,1\}$

包括NP-hard问题！

线性规划松弛化

最大独立集：

- $\max \sum_{v\in V} c_v x_v$
- $x_u + x_v \leq 1 \qquad \forall uv \in E$
- $x_v \in \{0,1\}$

松弛化 (relaxation)： $x_v \in \{0,1\}$ 松弛化为 $x_v \in [0,1]$。先找出一个分数解，然后再从分数解里面想办法找出一个整数解

最坏的情况下面，松弛化可能引入非常大的误差。

考虑完全图上的最大独立集，最优的分数解和整数解之间相差多少？

线性规划松驰化

最大二分图匹配：

- $\max \sum_{e\in E} c_e x_e$
- $\sum_{e\in\delta(v)} x_e \leq 1$ $\qquad \forall v \in V$
- $x_e \in \{0,1\}$ $\qquad \forall e \in E$

但是，松驰化对于很多多项式时间可解的组合优化问题，往往可以证明松驰化之后并没有误差

分数解也可以转化为整数解

LP因此被广泛认为是一个解决组合优化问题的“统一算法框架”

直观的解释：这些LP的polytope的顶点都在整数点上

LP的顶点

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

顶点可以有3种定义：

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.

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.**<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. $rank(A^{=})=n$

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$P := \{Ax \le b\}$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$x + y \in P, x - y \in P$

$$A^=(x + y) \le b^=$$

$$A^=(x - y) \le b^=$$

其中 $A^=x = b^=$

因此有 $A^=y \le 0, A^=y \ge 0$进而 $A^=y = 0$

LP的顶点

考虑$\max\langle c, x\rangle$，约束在$P \coloneqq \{Ax \le b\}$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

拓展：整数线性规划、松驰化、升维

最大二分图匹配：

- $\max \sum_{e \in E} c_e x_e$
- $\sum_{e \in \delta(v)} x_e \leq 1$ $\quad \forall v \in V$
- $x_e \in \{0,1\}$ $\quad \forall e \in E$

最大独立集：

- $\max \sum_{v \in V} c_v x_v$
- $x_u + x_v \leq 1$ $\quad \forall uv \in E$
- $x_v \in \{0,1\}$

松驰化 (relaxation)：$x_v \in \{0,1\}$ 松驰化为 $x_v \in [0,1]$，然后再找出一个整数解

升维 (extended formulation)

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