PKU-04833060
算法设计与分析『实验班』

第13讲 随机算法 (下)

罗国杰

gluo@pku.edu.cn

2025年春季学期

主要内容

- 泊松 （Poisson）试验
- 切诺夫（Chernoff）界
  - 关于独立随机变量的和 $X = \sum_{i=1}^{n} X_i$ 分布的尾概率的界
  - 尾概率 $Pr[X > (1+\delta)\mu]$、$Pr[X < (1-\delta)\mu]$ 以及 $Pr[|X-\mu| > c\mu]$，其中 $\mu = E[X]$
- Chernoff 界的应用
  - 负载均衡算法分析
  - 排列路由问题 和 随机无关路由算法
- 所有点对最短路径

本课内容主要出自

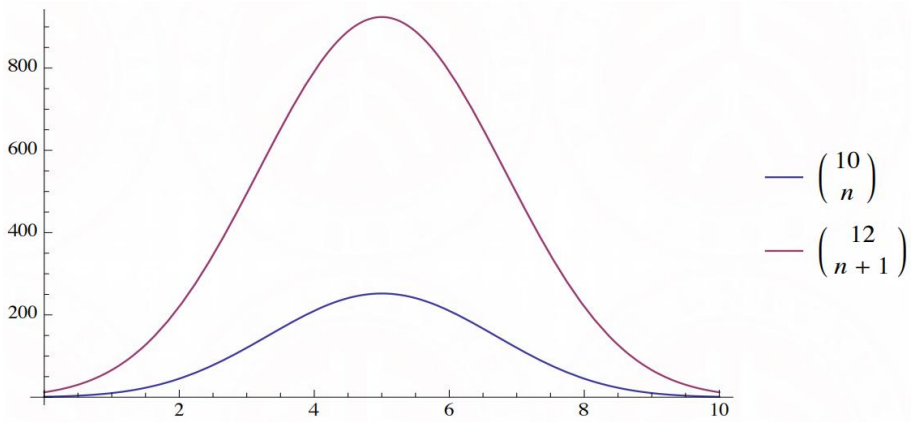
- “13.10 Load Balancing,” in J. Kleinberg and E. Tardos, Algorithm Design. Pearson Education, 2006.
- “4.2 Routing in a Parallel Computer” and “10.1 All-pairs Shortest Paths” in R. Motwani, P. Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
- Alistair Sinclair, "Lecture 13: Chernoff bounds + Randomized Routing", in Randomness & Computation, Berkeley CS 271, Fall 2011.

独立伯努利（Bernoulli）试验

- 独立伯努利（Bernoulli）试验
  - 令 $X_1, X_2, \cdots, X_n$ 是独立 Bernoulli 试验，即

$$\Pr[X_i = 1] = p \quad \wedge \quad \Pr[X_i = 0] = 1 - p,\ 1 \le i \le n$$

  - 令 $X = \sum_{i=1}^{n} X_i$
  - 则 X 称为具有二项分布。



泊松（Poisson）试验

- 泊松（Poisson）试验（更一般意义的独立随机试验）
  - 令 $X_1, X_2, \cdots, X_n$ 是独立硬币投掷，即

$$\Pr[X_i = 1] = p_i \quad \wedge \quad \Pr[X_i = 0] = 1 - p_i,\ 1 \le i \le n$$

  - 称这样的硬币投掷为 Poisson 试验，令

$$X = \sum_{i=1}^{n} X_i\text{，显然，}\mu = E[X] = \sum_{i=1}^{n} p_i$$

  - 考察 X 偏离其期望 μ 的两个相关问题：

$$\Pr[X > (1+\delta)\mu] < ?$$

$$\Pr[X > (1+\delta)\mu] < \epsilon \Rightarrow \delta > ?$$

切诺夫（Chernoff）界

定理 1 令$X_1, \ldots, X_n$为独立 Poisson 试验，即对于$1 \le i \le n$，有$\Pr[X_i = 1] = p_i$，其中$0 < p_i < 1$。对于$X = \sum_{i=1}^{n} X_i$，$\mu = \mathrm{E}[X] = \sum_{i=1}^{n} p_i$，以及$\forall \delta > 0$，

$$\Pr[X > (1+\delta)\mu] < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$$

证明：对于任意的正实数t，

$$\Pr[X > (1+\delta)\mu] = \Pr\left[e^{tX} > e^{t(1+\delta)\mu}\right] < \frac{\mathrm{E}[e^{tX}]}{e^{t(1+\delta)\mu}}$$

注意：此不等式是严格的，因为$0 < p_i < 1$。

定理1 证明（续1）

因为X_i是独立的，所以e^{tX_i}也彼此独立，于是

$$\mathbf{E}[e^{tX}] = \mathbf{E}\left[e^{t\left(\sum_{i=1}^{n} X_i\right)}\right] = \mathbf{E}\left[\prod_{i=1}^{n} e^{tX_i}\right] = \prod_{i=1}^{n} \mathbf{E}[e^{tX_i}]$$

考虑到

$$\Pr[e^{tX_i} = e^t] = \Pr[X_i = 1] = p_i$$

$$\Pr[e^{tX_i} = 1] = \Pr[X_i = 0] = 1 - p_i$$

$$\mathbf{E}[e^{tX_i}] = e^t \cdot p_i + 1 \cdot (1 - p_i) = (e^t - 1)p_i + 1 < e^{(e^t-1)p_i}$$

上式令$z = (e^t - 1)p_i$并利用$1 + z < e^z$关系可得。因此，

$$\Pr[X > (1+\delta)\mu] < \frac{\prod_{i=1}^{n} e^{(e^t-1)p_i}}{e^{t(1+\delta)\mu}} = \frac{e^{(e^t-1)\sum_{i=1}^{n} p_i}}{e^{t(1+\delta)\mu}} = \frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}$$

定理1 证明（续2）

由于对于所有的$t>0$，下式都成立

$$\Pr[X>(1+\delta)\mu]<\frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}$$

对右式关于t求导得

$$\frac{d}{dt}\left(\frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}\right)=\frac{d}{dt}\left(e^{\left((e^t-1)-t(1+\delta)\right)\mu}\right)$$

$$=\left(e^{\left((e^t-1)-t(1+\delta)\right)\mu}\right)\cdot\mu\cdot\left(e^t-(1+\delta)\right)$$

令求导式等于0求得

$$e^t-(1+\delta)=0\Rightarrow t=\ln(1+\delta)$$

时，求得最紧的界。

切诺夫（Chernoff）界

定理 2 令$X_1, \ldots, X_n$为独立 Poisson 试验，即对于$1 \le i \le n$，有$\Pr[X_i = 1] = p_i$，其中$0 < p_i < 1$。对于$X = \sum_{i=1}^{n} X_i$，$\mu = \mathrm{E}[X] = \sum_{i=1}^{n} p_i$，以及$\forall \delta, 0 < \delta \le 1$，

$$\Pr[X < (1-\delta)\mu] < \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu} < e^{-\frac{\delta^2}{2}\mu}$$

定理2 证明

证明：对于任意的正实数t，

$$\Pr[X<(1-\delta)\mu]=\Pr\left[e^{-tX}>e^{-t(1-\delta)\mu}\right]<\frac{\mathrm{E}[e^{-tX}]}{e^{-t(1-\delta)\mu}}<\frac{e^{(e^{-t}-1)u}}{e^{-t(1-\delta)\mu}}$$

令$t=\ln\left(\frac{1}{1-\delta}\right)$，得

$$\Pr[X<(1-\delta)\mu]<\left[\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right]^{\mu}$$

利用$\delta\in(0,1]$，

$$(1-\delta)^{(1-\delta)}>e^{-\delta+\frac{\delta^2}{2}}$$

得到

$$\Pr[X<(1-\delta)\mu]<e^{-\frac{\mu\delta^2}{2}}$$

切诺夫（Chernoff）界

定理 3 令$X_1, \ldots, X_n$为独立 Poisson 试验，即对于$1 \le i \le n$，有$\Pr[X_i = 1] = p_i$，其中$0 < p_i < 1$。对于$X = \sum_{i=1}^{n} X_i$，$\mu = \mathrm{E}[X] = \sum_{i=1}^{n} p_i$，以及$\forall c > 0$，

$$\Pr[|X - \mu| > c\mu] < 2e^{-\min\left\{\frac{c^2}{4}, \frac{c}{2}\right\}\mu}$$

定理3 证明

证明：当$1 \geq c > 0$时，$\Pr[X < (1-c)\mu] \leq e^{-\frac{c^2}{2}\mu}$

当$c > 1$时，$\Pr[X < (1-c)\mu < 0] = 0 \leq e^{-\frac{c^2}{2}\mu}$，于是

$$\Pr[\mu - X > c\mu] = \Pr[X < (1-c)\mu] \leq e^{-\frac{c^2}{2}\mu} \leq e^{-\frac{c^2}{4}\mu}$$

} $Pr[\mu - X > c\mu]$

当$c \geq 2$时，$\Pr[X - \mu > c\mu] < \left[\frac{e^c}{(1+c)^{(1+c)}}\right]^{\mu} \leq e^{-\frac{c}{2}\mu}$

当$c \leq 2$时，$\Pr[X - \mu > c\mu] < \left[\frac{e^c}{(1+c)^{(1+c)}}\right]^{\mu} \leq e^{-\frac{c^2}{4}\mu}$

} $Pr[X - \mu > c\mu]$

$$\begin{aligned}
&\Pr[|X - \mu| > c\mu] \\
&= \Pr[\mu - X > c\mu] + \Pr[X - \mu > c\mu] \\
&< 2e^{-\min\left\{\frac{c^2}{4}, \frac{c}{2}\right\}\mu}
\end{aligned}$$

Load Balancing

- Load balancing. System in which m jobs arrive in a stream and need to be processed immediately on n identical processors. Find an assignment that balances the workload across processors.
- Centralized controller. Assign jobs in round-robin manner. Each processor receives at most $\lceil m/n \rceil$ jobs.
- Decentralized controller. Assign jobs to processors uniformly at random. How likely is it that some processor is assigned “too many” jobs?

Load Balancing: Analysis (m=n)

- Let X_i = number of jobs assigned to processor i
- Let $Y_{ij} = 1$ if job j assigned to processor i, and 0 otherwise
- We have $E[Y_{ij}] = 1/n$
- Thus, $X_i = \sum_j Y_{ij}$, and $\mu = E[X_i] = 1$
- Applying Chernoff bounds with $\delta = c - 1$ yields $Pr[X_i > c] < e^{c-1}/c^c$
- Let $\gamma(n)$ be number x such that $x^x = n$, and choose $c = e\gamma(n)$

$$Pr[X_i > c] < e^{c-1}/c^c < (e/c)^c = (1/\gamma(n))^{e\gamma(n)} < (1/\gamma(n))^{2\gamma(n)} = 1/n^2$$

- Union bound ⇒ with probability ≥ 1-1/n no processor receives more than $e\gamma(n) = \Theta(\log n/\log\log n)$ jobs

Load Balancing: Many Jobs

- Theorem. Suppose the number of jobs $m = 16n\ln n$. Then on average, each of the n processors handles $\mu = 16\ln n$ jobs. With high probability, every processor will have between half and twice the average load.
- Proof.
  - Let X_i, Y_{ij} be as before.
  - Applying Chernoff bound with δ = 1 yields $Pr[X_i > 2\mu] < (e/4)^{16\ln n} < (1/e)^{\ln n} = 1/n^2$
  - Applying Chernoff bound with δ = 1/2 yields $Pr[X_i < \frac{1}{2}\mu] < e^{-\frac{1}{2}(\frac{1}{2})^2\ 16\ln n} = 1/n^2$
  - Union bound ⇒ every processor has load between half and twice the average with probability ≥ 1-2/n

并行计算机的路由问题

- 考虑有 ***N*** 个处理器，标号从 **1** 到 ***N***，并行处理器的网络抽象为一个图。
- 直接互联的处理器间可以直接通信；
- 不直接互联的处理器间的通信必须通过其他处理器转发。
- 假设每个处理器都可以在单个同步周期（一步）内向所有直接互联的处理器分别发送一个单位的消息。

- 考虑 排列路由问题 及其 无关路由算法类
- 利用 **Chernoff** 界可证明：该算法类中，任何确定性算法都不会比随机算法更好

排列路由（Permutation Routing）问题

- 设开始时，每个处理器 i 均有一个消息包 v_i 将要发给处理器 $\pi(i)$。其中 $\pi(i)$（$1 \le i \le N$）构成 $\{1 \ldots N\}$ 的一个排列，即每个处理器恰好是一个消息包的目的地。
- 包的路由，即包从处理器 i 至处理器 $\pi(i)$ 所经过的节点或边的序列。
- 排列路由问题要为包确定一条路由。多个等待在同一条边上发送的消息包通过某种排队机制按顺序处理。

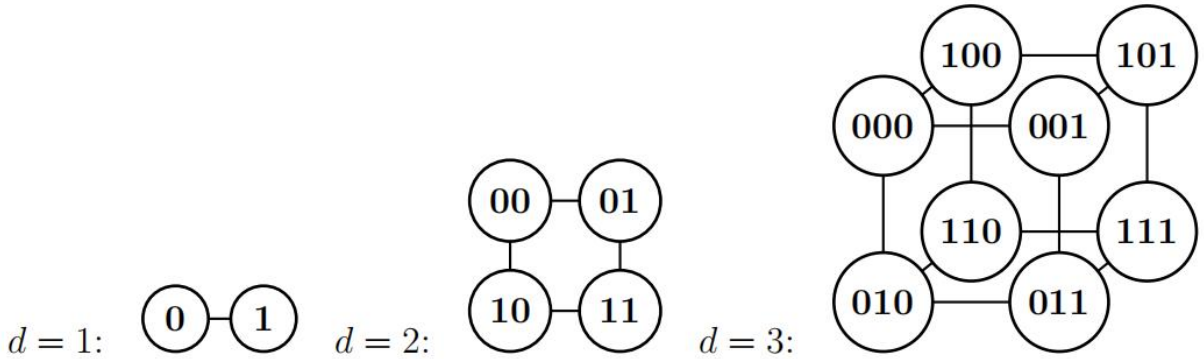
无关路由（Oblivous Routing）算法类

- 无关路由：消息 v_i 采用的路由仅与它的目的地相关，而与其它任意消息的目的地无关。
  - [Val82] L. G. Valiant, “A Scheme for Fast Parallel Communication,” SIAM J. Comput., vol. 11, no. 2, pp. 350–361, May 1982.

- 定理：排列路由问题的任意确定性无关路由算法，对于由出度为 $n = \log N$ 的 N 个结点构成的超立方体网络，总存在一个排列路由的实例至少需要 $\Omega(\sqrt{N}/n)$ 步
  - [KKT91] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. Tight bounds for oblivious routing in the hypercube. Theory of Computing Systems, 24(1):223–232, 1991.

超立方体网络（1/2）

- n 维超立方体网络有 $N=2^n$ 个结点
- 想象正方形（2维）或正方体（3维）的n维版本



超立方体网络（2/2）

- 考虑一个布尔超立方体并行处理器网络 $G = (V, E)$：

$$V = \{i \in \mathbb{N} \mid 0 \le i < N, N = 2^n\}$$

设$(i_0, i_1, \ldots, i_{n-1}) \in \{0, 1\}^n$为$i$的二进制表示，即

$$i = \sum_{k=0}^{n-1} i_k 2^k$$

$$E = \{(i, j) \mid \exists k, \forall t \neq k, i_t = j_t, i_k \neq j_k\}$$

边只存在于结点编号二进制表示仅有一位不同的结点间

超立方体网络的 Bit-Fixing 路由（一种无关路由）

- 从 i 发送消息 v_i 到结点 $\pi(i)$ 的路由算法为：
  - 从左到右扫描 $\pi(i)$ 的二进制位，与 v_i 当前所在结点 j 的地址比较
  - 找到当前地址与 $\pi(i)$ 不同的最左边的二进制位对应的边 $e(j, \pi(i))$
  - 从 $e(j, \pi(i))$ 将消息 v_i 转发出结点 j 。
- 若 i = 1011、$\pi(i)$ = (0000)，则 v_i 所经历的结点序列为：
  - 第1次转发 1011 → 0011 ： e(j=1011, π(i)=0000)
    第2次转发 0011 → 0001 ： e(j=0011, π(i)=0000)
    第3次转发 0001 → 0000 ： e(j=0001, π(i)=0000)
- 最坏情况步数的下界是 $\Omega(\sqrt{N}/n)$

（提示：对于 π(ab)=ba 形式的排列，考虑 xx 形式结点的消息传递步数）

两阶段的随机无关路由算法

对于每条消息 v_i，独立的执行下面两个阶段。

- 阶段1：从 $\{1, \ldots, N\}$ 中随机选择中间目标 $\sigma(i)$，将 v_i 传输到结点 $\sigma(i)$。
- 阶段2：等待至第 $7n$ 步开始，将 v_i 从 $\sigma(i)$ 传输到目标结点 $\pi(i)$。

（阶段1和阶段2均采用 Bit-Fixing 路由算法）

（拥塞结点的排队规则：不同时相遇，先进先出；同时相遇，可以任意顺序转发）

定理：上述随机路由算法以至少 $1-(1/N)$ 的概率，使每个包在 $14n$ 或更少的步数内达到它的目标。

（两阶段是对称的，只需证明任一阶段大概率在 $7n$ 步内完成）

随机无关路由算法的运行时间分析

- 即分析每个 v_i 到达目的结点需要花费的步数。
- 先分析 v_i 在“阶段1”中的路由过程。
  所需步数由两部分构成：
  - 路线 ρ_i 的长度（至多为 n）；
  - 在路线 ρ_i 的中间结点排队（延迟）的步数（至多为 $|S_i|$，定义见下一页）。

随机无关路由算法：阶段1中消息包的延迟

- 关于延迟上界的引理
  - 令消息 v_i 的路由为有向边的序列 $\rho_i = \{e_1, e_2, \ldots, e_k\}$ 。
  - 令 S_i 为除 v_i 外、经过 ρ_i 任意边的消息集合：$S_i = \{v_j \mid v_j \neq v_i \text{ and } \rho_j \cap \rho_i \neq \Phi\}$，即消息 v_j 的路由至少经过 $\{e_1, e_2, \ldots, e_k\}$ 中的某条边
  - 那么，消息 v_i 的延迟至多为 $|S_i|$ 。

- 证明：如果某时刻 v_j 导致 v_i 等待，v_i 给 v_j 发放一枚 token；如果某时刻 v_k ($k \neq i$) 导致带 token 的 v_j 等待，v_j 将 token 转移至 v_k。可证明 S_i 的每条消息在任意时刻至多携带一枚 token，也即导致 v_i 等待的次数最多 $|S_i|$ 次。

随机无关路由算法：关于延迟的期望 和 界

- 定义随机变量 H_{ij}

$$H_{ij} = \begin{cases} 1 & \text{if } \rho_i \cap \rho_j \neq \Phi \\ 0 & \text{else} \end{cases}$$

- 则消息 v_i 的总延迟至多为

$$|S_i| \leq \sum_{j=1}^{N} H_{ij}$$

- H_{ij} ($i \neq j$) 是独立的 Poisson 试验，可以用 Chernoff 界估计
- 但首先考虑延迟的期望（或它的界） $\mathbf{E}\left[\sum_{j=1}^{N} H_{ij}\right]$

随机无关路由算法：估计延迟期望的界

- 考虑随机变量 $T(e)$ 表示通过边 e 的路由的数目。
- 对于任意固定的路由线路 $\rho_i = \{e_1, e_2, \ldots, e_k\}$，有

$$\mathbf{E}\left[\sum_{j=1}^{N} H_{ij}\right] \leq \mathbf{E}\left[\sum_{l=1}^{k} T(e_l)\right] = \sum_{l=1}^{k} \mathbf{E}[T(e_l)]$$

- 由于超立方体的对称性：$\forall l, m, \mathbf{E}[T(e_l)] = \mathbf{E}[T(e_m)]$
- 路由 ρ_i 边数的期望为 n/2，总共 N 条路由，总共 N n 条有向边，每条边被通过的路由线路数目的期望：E[T(e)] = N * n/2 / (N n) = 1/2

随机无关路由算法：延迟的 Chernoff 界（1/2）

- 于是 $\mathbf{E}[T(e_l)] = \frac{1}{2} \Rightarrow \mathbf{E}\left[\sum_{j=1}^{N} H_{ij}\right] \leq \frac{k}{2} \leq \frac{n}{2}$

- 由 Chernoff 界（定理1）可得：$\mathbf{Pr}[X \geq R] \leq 2^{-R}, \mathbf{for}\ R \geq 6E[X]$
  - 令 **R = (1+δ) μ**，

$$\begin{aligned}
\text{Prob}[X \geq (1+\delta)\cdot\mu] &\leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu} \\
&\leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)\cdot\mu} \\
&\leq \left(\frac{e}{6}\right)^{R} \\
&\leq 2^{-R}.
\end{aligned}$$

随机无关路由算法：延迟的 Chernoff 界（2/2）

- 由 $\mathbf{Pr}[X \geq R] \leq 2^{-R}, \mathbf{for}\ R \geq 6E[X]$
  - 可证明单条消息延迟超过 $6n$ 的概率 $\leq 2^{-6n}$
  - 存在消息延迟超过 6n 的概率（union bound）$\leq N \times 2^{-6n} = 2^n \times 2^{-6n} = 2^{-5n}$
- 定理：在阶段1中，以至少 $1-2^{-5n}$ 的概率，每个包在 $7n$ 或更少的步数内达到它的中间目标。
  - 步数 = 路由长度 + 延迟 超过 $n + 6n = 7n$ 的概率 $\leq 2^{-5n}$
- 定理：随机算法以至少 $1 - 1/N$ 的概率，每个包在 $14n$ 或更少的步数内到达终点。
  - 失败概率 $\leq 2 * 2^{-5n} < 1/N$
- 对比：确定性算法步数的下界是 $\Omega(\sqrt{N}/n)$

所有点对最短路径（APSP问题）

- 问题描述
  - 给定一个无向无权连通图$G(V,E)$，求其中所有点对之间的最短路径。
- 普通解法
- 随机算法
  - 第一阶段求解APD问题（确定性）
  - 第二阶段求解APSP问题（随机）

所有点对最短路径：普通解法

- 最简单直接的想法，我们可以利用 Floyd 算法来对该问题进行求解。算法的复杂度是$O(n^3)$的。
- 或者，对于每个顶点 i，我们可以用最短路算法或者广搜方法求出其到达其他任何一个顶点的最短路。对所有顶点使用该算法后即可获得所有点对之间的最短距离。由于最短路算法是 $O(n^2)$ 或者 $O(m\log n)$ 的，而在稠密情况下 $m = O(n^2)$。故总的算法代价仍然是 $O(n^3)$ 的。

- 通过上述的分析，用普通算法解决该问题，时间复杂度为 $O(n^3)$ 。
- 下面将介绍一种带有随机机制的算法，使得时间复杂度降为 $O(MM(n)\cdot\log^2 n)$。

所有点对最短路径长度（APD问题）

- 为了更简单的入手分析这个问题，我们首先先求出任意两点之间的最短距离，这个问题我们称之为所有点对之间最短路径长度问题（all-pairs distances，APD）。
- 这一问题其实通过任意一个最短路算法都可以解决，但是为了之后将问题推广到求最短路径，我们介绍一种比较特殊的方法来解决APD问题。

重新定义APD问题

- 设 A 为图 G 的邻接矩阵，则 $A_{ij} = 1$ 当且仅当 i、j 在图 G 中相邻。
- 设矩阵 $C = A^n$，则 C_{ij} 等于图 G 中在 i、j 之间距离为 n 的通路数量。
- 图的直径：图中任意两点间最短路径的最大值。
- 设 G 是一个无向无权连通图，我们将图 G 中所有距离为1或2的点对之间加一条边，得到新图 G’，称 G’ 为 G 的平方。
- 我们定义 G 的邻接矩阵为 A，距离矩阵为 D；G’ 的邻接矩阵为 A’，距离矩阵为 D’。

- 第一阶段求解 APD 问题的策略是根据 G 的邻接矩阵 A 求出 G 的距离矩阵 D。
- 在之后的算法过程中我们会经常用到矩阵 A 的平方，定义其为 Z，即 Z=AA。

求解APD的确定性算法（稍后分析）

- 算法输入：图 G 的邻接矩阵 A
- 算法输出：图 G 的距离矩阵 D

1. 计算 Z=AA
2. 计算 A’，即得到图 G 的平方 G’
3. 如果 G’ 是完全图，则利用 D=2A’-A 得到 D，返回。
4. 否则，递归调用该算法，利用 A’ 计算得到 D’。
5. 根据 D'_{ij} 判断 D_{ij} 的奇偶性，然后计算得到 D，返回。

求解APD：矩阵 A’ 的计算

- **定理**：在图 G’ 中，两点 i 和 j 相邻当且仅当 i 和 j 在图 G 中的距离为1或2。
- 根据这一性质，我们就可以利用 A 和 Z 计算 A’ 了，具体方法为：

$$A'_{ij} = 1 \quad \text{当且仅当} \quad i \neq j \text{ 且 } Z_{ij} \text{ 和 } A_{ij} \text{ 不都为} 0$$

- 下面我们考虑如何利用 A’ 来计算 D。

求解APD：矩阵 D 的计算

- 注意到一个性质：当且仅当 G 的直径不超过 2 时，G’ 是完全图，A’ 是除对角元素为 0 外其他所有元素为 1 的矩阵。这种情况我们称之为终止情况。
- 此时，D 可以通过 A 和 A’ 直接计算得到，公式为：

$$D=2A'-A$$

因为 $A_{ij} = 1 \Leftrightarrow$ (i,j) 有距离为1的路径

$A'_{ij} = 1 \Leftrightarrow$ (i,j) 有距离为1或2的路径

求解APD：矩阵 D 的计算

- 对于 G 的直径大于 3，即 G’ 不是完全图的情况，我们称之为一般情况。在一般情况下，可以利用 D’ 来计算 D。公式如下：
  - 如果 D_{ij} 是偶数，则 $D_{ij} = 2D'_{ij}$
  - 如果 D_{ij} 是奇数，则 $D_{ij} = 2D'_{ij} - 1$（因为 $D_{ij} - 1 = 2(D'_{ij} - 1)$）
  - 容易证明这种方法的正确性。
- 于是，现在问题被转化为求无向无权连通图 G’ 的距离矩阵 D’，这个问题可以递归求解。因为每次对图求平方时，都会使任意两点间的距离缩小近一半，所以经过若干次求平方，图一定会变为一个完全图，这就是上页中提到的递归算法的终止情况，也是递归的终点。

求解APD：思路整理

- 算法输入：图G的邻接矩阵 A
- 算法输出：图G的距离矩阵 D

1. 计算 Z=AA
2. 计算 A’，即得到图 G 的平方 G’
3. 如果 G’ 是完全图，则利用 D=2A’-A 得到 D，返回。
4. 否则，递归调用该算法，利用 A’ 计算得到 D’，然后利用上页中的公式用 D’ 来计算D。

- 现在的问题的关键就在于：如何确定D_{ij}的奇偶性

求解APD：如何确定 D_{ij} 的奇偶性

- **引理**：对于图 G 中任意一对不同的顶点 i 和 j，对于 i 的任意邻居 k，有 $D_{ij} - 1 \leq D_{kj} \leq D_{ij} + 1$；顶点 i 存在邻居 k 使得 $D_{kj} = D_{ij} - 1$。
- 由引理继而得到下一个性质：
- **性质**：
  - 如果 D_{ij} 为偶数，则对于顶点 i 的每一个邻居 k 来说，$D'_{kj} \geq D'_{ij}$；
  - 如果 D_{ij} 为奇数，则对于顶点 i 的每一个邻居 k 来说，$D'_{kj} \leq D'_{ij}$。

  （可根据 D_{kj} 的奇偶性分情况检验上述性质的正确性）
- 对于 i 的每个邻居，对上述等式做累加，通过比较两者的大小就可以判断 D_{ij} 的奇偶性。
- 至此，APD 问题已经完全解决。

求解APD：最终算法

- 算法输入：图 G 的邻接矩阵 A
- 算法输出：图 G 的距离矩阵 D

1. 计算 Z=AA
2. 计算 A’，即得到图 G 的平方 G’
3. 如果 G’ 是完全图，则利用 D=2A’-A 得到 D，返回。
4. 否则，递归调用该算法计算，利用 A’ 计算得到 D’。
5. 根据 D'_{ij} 判断 D_{ij} 的奇偶性，然后计算得到 D，返回。

APD 问题复杂度分析

- 设 ***MM*(*n*)** 表示两个 ***n*** 阶方阵相乘的时间复杂度。
  - ~~例如 $O(n^{2.37286})$ [Alman, Williams, SODA'21]~~
  - 例如 $\boldsymbol{O(n^{2.371552})}$ [Williams, Xu, Xu, Zhou, SODA'24]
- 假设图 ***G*** 的直径为 ***d***，则图 ***G'*** 的直径为 **[*d*/2]**。令 ***T*(*n*,*d*)** 表示 APD 算法在面对一个有 ***n*** 个顶点，直径为 ***d*** 的图作为输入时的运行时间。
- 当 ***d*=2** 时，$T(n,d)=MM(n)+O(n^2)$
- 当 ***d*>2** 时，$T(n,d)=2MM(n)+T(n,[d/2])+O(n^2)$
- 而递归的深度至多为 **log*n***，故算法的总复杂度为 $\boldsymbol{O(MM(n)\cdot \log n)}$。

求解 APSP 问题

- 下面我们在已经求得所有点之间最短距离 APD 的基础上，求所有点之间的最短路径。我们称之为无权图的所有点之间最短路径问题（all-pairs shortest paths problem, APSP）。
- 我们将在解决 APSP 问题时使用随机机制。

求解 APSP 问题：后继矩阵

- 对于图 G，设矩阵 S，其中 S_{ij} 是由 i 到 j 的最短路径上顶点 i 后继邻居的标号。称 S 是图 G 的后继矩阵。
- 如果我们得到了图 G 的后继矩阵 S，则对于任意两点 i 和 j，我们就可以通过每次从后继矩阵获取后继邻居的方法，一步步的得到他们之间的最短路径。
- 于是求解 APSP 的问题等效于根据图 G 的邻接矩阵 A 和距离矩阵 D 来求它的后继矩阵 S。

求解 APSP 问题：布尔矩阵相乘的“证据”

- 假设 ***A*** 和 ***B*** 是 ***n*n*** 的布尔矩阵，***P=AB*** 是他们在布尔矩阵乘法下的乘积。
- 元素 $\boldsymbol{P_{ij}}$ 的一个证据是指一个标号 $\boldsymbol{k} \in \{\boldsymbol{1}, \ldots, \boldsymbol{n}\}$，使得$\boldsymbol{A_{ik}} = \boldsymbol{B_{kj}} = \boldsymbol{1}$。
- 因为这里使用的乘法是布尔乘法，故
- P_{ij} 存在至少一个证据 当且仅当 $P_{ij} = 1$。
  - 当然，***P*** 的每个元素都可能拥有多个“证据”。
- 假设 ***C=AB*** 是 ***A*** 和 ***B*** 的整数乘积（非布尔乘积），则 $\boldsymbol{C_{ij}}$ 表示了 $\boldsymbol{P_{ij}}$ 的证据数。

求解 APSP 问题：“证据”的图论意义

- 假设 A 是一个图 G 的邻接矩阵，$B = A^x$，$X = A^y$，$P = A^x A^y = BX$。
  - 此处的乘法全部为布尔乘法。
- 则 $B_{ik} = 1$ 表示从 i 到 k 存在长度为 x 的路径，
- 且 $X_{kj} = 1$ 表示从 k 到 j 存在长度为 y 的路径。
- $P_{ij} = 1$ 表示存在一条从 i 到 j 的长度为 $x+y$ 的路径，
- 而 P_{ij} 如果存在一个“证据” k，则说明 $B_{ik} = 1$ 且 $X_{kj} = 1$，
  - 其意义为存在一条从 i 到 j 且经过 k 的长度 $x+y$ 的路径。
  - 进一步可知在这条路径上从 i 到 k 的距离为 x，从 k 到 j 的距离为 y。
- 这就是“证据”在图论上的意义。

求解 APSP 问题：“证据”与后继矩阵 S 的关系

- 对于两个节点 i 和 j，假设他们的距离为 d。
- 令 A 为图 G 的邻接矩阵，矩阵 $B = A^{d-1}$，$P = AB = A^{d}$，乘法均为布尔乘法。
- 显然 $P_{ij} = 1$，如果它有一个证据 k，根据上页的分析，就可以说明存在一条从 i 到 j 经过 k 的距离为 d 的路径，也就是从 i 到 j 的最短路。而且，i 到 k 的距离为 1，即 k 是 i 的邻居。
- 根据上述性质，这个 k 就是一个符合要求的 S_{ij}。
- 只要我们对任意的 i 和 j 都找出 P_{ij} 的一个证据，我们就能得到后继矩阵。

求解 APSP 问题：思路整理

- 算法输入：图 ***G*** 的邻接矩阵 ***A*** 和距离矩阵 ***D***
- 算法输出：图 ***G*** 的后继矩阵 ***S***
- 1. 预处理，得到矩阵 A、A^2、…、A^n（布尔乘法）
- 2. 对于任意点对 ***i*** 和 ***j***：
  - 从距离矩阵 ***D*** 中得到 ***i*** 至 ***j*** 的距离 ***d***
  - 得到矩阵 $B = A^{d-1}$
  - 得到矩阵 ***P = AB***
  - 找到 P_{ij} 的一个证据 k，$S_{ij} = k$
- 3. 返回 ***S***

- 现在问题的关键为： 如何找到 P_{ij} 的一个证据。

求解 APSP 问题：如何找到 P_{ij} 的一个证据（1/3）

设 $P = AB$。如果 P_{ij} 只含有一个证据：

- 定义矩阵 T，$T_{ik} = kA_{ik}$。
- 设矩阵 $W = TB$（整数乘法），则 W_{ij} 即为 P_{ij} 的证据。

求解 APSP 问题：如何找到 P_{ij} 的一个证据（2/3）

- 当然，我们不能保证每一个 P_{ij} 都只有一个证据。但是我们可以利用随机机制来保证 P 中充分多的元素具有这一性质。
- 假设 P_{ij} 的证据个数为 w，这个值可以从矩阵 C 中直接得到。如果 $w=1$，则可以按照上页的方法直接求得。
- w 大于 1 时：取一个整数 r，r 满足 $\frac{n}{2} \leq wr \leq n$。我们从 1~$n$ 这 n 个数中随机挑选 r 个数构成集合 R。可以证明，R 中只包含 P_{ij} 的一个证据的概率 $\mathbf{Pr} \geq \frac{1}{2e}$。

$$
\begin{aligned}
\frac{C_w^1 C_{n-w}^{r-1}}{C_n^r} &= w\frac{r!}{(r-1)!}\frac{(n-w)!}{n!}\frac{(n-r)!}{(n-w-r+1)!} \\
&= wr(\prod_{i=0}^{w-1}\frac{1}{n-i})(\prod_{j=0}^{w-2}(n-r-j)) \\
&= \frac{wr}{n}(\prod_{j=0}^{w-2}\frac{n-r-j}{n-1-j}) \\
&\geq \frac{wr}{n}(\prod_{j=0}^{w-2}\frac{n-r-j-(w-j-1)}{n-1-j-(w-j-1)}) \\
&= \frac{wr}{n}(\prod_{j=0}^{w-2}\frac{n-w-(r-1)}{n-w}) \\
&= \frac{wr}{n}(1-\frac{r-1}{n-w})^{w-1} \\
&\geq \frac{1}{2}(1-\frac{1}{w})^{w-1}
\end{aligned}
$$

求解 APSP 问题：如何找到 P_{ij} 的一个证据（3/3）

- 假设集合 R 包含元素 P_{ij} 的唯一证据，我们可以用类似只包含一个证据的方法来求得这个证据。
- 设 R 表示一个向量，如果 k 在集合 R 中，则 $R_k = 1$，否则 $R_k = 0$。
- 设邻接矩阵 A 和 $B = A^{d-1}$ 的含义同算法中的定义。设矩阵 A^R 满足 $A^R_{ik} = kR_kA_{ik}$，矩阵 B^R 满足 $B^R_{kj} = R_kB_{kj}$。这样的处理与之前的处理唯一的差别就在于只保留了两个矩阵中 R_k 为 1 的列。
- 设矩阵 $P^R = A^RB^R$，则 $P^R{}_{ij}$ 即为 P_{ij} 的一个证据。

可证明算法期望运行时间是 $O(MM(n)log^2n)$

Algorithm BPWM:

Input: Two $n \times n$ 0-1 matrices A and B.

Output: Witness matrix W for the Boolean matrix $P = AB$.

1. $W \leftarrow -AB$.
2. **for** $t = 0, \ldots, \lceil \log n \rceil$ **do**
   - 2.1. $r \leftarrow 2^t$.
   - 2.2. **repeat** $\lceil 3.77 \log n \rceil$ **times**
     - 2.2.1. **choose** random $R \subseteq \{1, \ldots, n\}$ with $|R| = r$.
     - 2.2.2. **compute** A^R and B^R.
     - 2.2.3. $Z \leftarrow A^R B^R$.
     - 2.2.4. **for all** (i, j) **do**
       if $W_{ij} < 0$ and Z_{ij} is witness **then** $W_{ij} \leftarrow Z_{ij}$.
3. **for all** (i, j) **do**
   if $W_{ij} < 0$ **then** find witness W_{ij} by brute force.

求解 APSP：最终算法

Algorithm APSP:

Input: An $n \times n$ adjacency matrix A for a graph G.

Output: The successor matrix S for G.

1. **compute** the distance matrix $D = \mathbf{APD}(A)$.
2. **for** $s = \{0, 1, 2\}$ **do**

    2.1. **compute** 0-1 matrix $D^{(s)}$ such that $D^{(s)}_{kj} = 1$ if and only if $D_{kj} + 1 \equiv s \pmod 3$.

    2.2. **compute** the witness matrix $W^{(s)} = \mathbf{BPWM}(A, D^{(s)})$.
3. **compute** successor matrix S such that $S_{ij} = W_{ij}^{(D_{ij} \bmod 3)}$.

for any k such that Akj = 1 and Dkj = Dij - 1 (mod 3) is a valid candidate of Sij; avoid trying all d's=O(n), reducing from n down to 3

本课小结

- 切诺夫（Chernoff）界用于评估偏离均值的概率。
- 只能适用于独立Poisson试验。
- 可以根据期望的概率评估可能的偏离值。
- 随机算法很多时候比确定性算法性质更好。

- 本课内容主要出自
  - “13.10 Load Balancing,” in J. Kleinberg and E. Tardos, Algorithm Design. Pearson Education, 2006.
  - “4.2 Routing in a Parallel Computer” and “10.1 All-pairs Shortest Paths” in R. Motwani, P. Raghavan, “Randomized Algorithms”, Cambridge University Press, 1995.
  - Alistair Sinclair, "Lecture 13: Chernoff bounds + Randomized Routing", in Randomness & Computation, Berkeley CS 271, Fall 2011.