

Lecture 13: 次线性算法：平均距离

2025.4.8

Lecturer: 丁虎

Scribe: 王运韬

1 平均距离问题

在几何图(边权 d 可以看作一个度量) (V, d) 中, 时常需要计算平均距离 $\sum_{p < q \in V} d(p, q) / \binom{|V|}{2}$. Indyk [1] 提出了如下的次线性算法:

1. 独立同分布地以概率 $\frac{s}{m}$ 采样每一条边, 得到一个期望大小为 s 的边合 S (n, m 分别是顶点数和边数, $s := an$).
2. 计算这些边的平均权重。

由 Markov 不等式, 该随机算法的时间复杂度以很高概率为 $O(s)$, 主要用于计算算法的第 2 步。我们的目标是, 提出误差为 δ 的次线性算法, 这就需要我们设计 a 的取值。

令 Δ 为这个图的最大边权。不失一般性, 假设最小的边权为 1 (可以让全部的边权除以之, 最后得出结果再乘回来)。对于 $0 < \epsilon < \delta$, 我们取 $c = 1 + \epsilon$, $I_i = [c^i, c^{(i+1)})$ (这里的上标是指数)。我们按照边权在哪个区间来定量分析: n_i 定义为边权落在 I_i 的边的个数, s_i 定义为 S 中边权在 I_i 的边的个数, $\tilde{A} = \sum_i c^i n_i$, $A' = \sum_{e \in S} d(e)$, $\tilde{A}' = \frac{m}{s} \sum_i c^i s_i$. 换句话说, 我们用区间的端点取值去逼近真实的平均距离, 从而有 $A = (1 \pm \epsilon)\tilde{A}$, $A' = (1 \pm \epsilon)\tilde{A}'$. 故而, 为了得出平均距离, 只需证明 \tilde{A}' 能近似拟合 \tilde{A} . 注意到 $\tilde{A} = \mathbb{E}\tilde{A}'$, 由切比雪夫不等式, 为逼近 A 我们只需让方差 $D^2(\tilde{A}')$ 尽可能小。

回顾概率论中有指示函数 $\mathbb{1}_{\mathcal{X}}(x) = \begin{cases} 1 & \text{如果 } x \in \mathcal{X}, \\ 0 & \text{如果 } x \notin \mathcal{X} \end{cases}$. 从而有 $s_i = \sum_{e: d(e) \in I_i} \mathbb{1}_{e \in S}$. 由于每条边的采样是独立的, 且每条边仅可能落在一个区间 I_i 内, 可知 $\{s_i : i \in \mathbb{N}_+\}$ 是一组独

立随机变量。因此，

$$\begin{aligned}
D^2(\tilde{A}') &= D^2\left(\frac{m}{s} \sum_i c^i s_i\right) && \text{(定义)} \\
&= \frac{m^2}{s^2} D^2\left(\sum_i c^i s_i\right) && \text{(提取系数)} \\
&= \frac{m^2}{s^2} \sum_i D^2(c^i s_i) && \text{(独立变量的和的方差)} \\
&= \frac{m^2}{s^2} \sum_i c^{2i} D^2\left(\sum_{e:d(e) \in I_i} \mathbf{1}_{e \in S}\right) \\
&= \frac{m^2}{s^2} \sum_i c^{2i} \sum_{e:d(e) \in I_i} D^2(\mathbf{1}_{e \in S}) && \text{(同样是独立变量的和的方差)} \\
&\leq \frac{m^2}{s^2} \sum_i c^{2i} \sum_{e:d(e) \in I_i} \mathbb{E}(\mathbf{1}_{e \in S}^2) && \text{(对任何随机变量 } X, D^2(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2) \\
&= \frac{m^2}{s^2} \sum_i c^{2i} \sum_{e:d(e) \in I_i} \Pr(e \in S) && \text{(对任何事件 } \mathcal{X}, \text{ 恒有 } \mathbb{E}\mathbf{1}_{\mathcal{X}} = \Pr(X)) \\
&= \frac{m^2}{s^2} \sum_i c^{2i} n_i \frac{s}{m} = \frac{m}{s} \sum_i c^{2i} n_i
\end{aligned}$$

现在，可以应用切比雪夫不等式，得到

$$\begin{aligned}
\Pr(|\tilde{A}' - E[\tilde{A}']| \geq \epsilon \cdot E[\tilde{A}']) &\leq \frac{1}{\frac{\epsilon^2 E^2[\tilde{A}']}{D^2[\tilde{A}']}} \\
&= \frac{D^2[\tilde{A}']}{\epsilon^2 E^2[\tilde{A}']} \\
&= \frac{1}{\epsilon^2} \frac{m}{s} F,
\end{aligned} \tag{1}$$

上式的 $F = \frac{\sum c^{2i} n_i}{\sum c^{2i} n_i^2}$. 这是源于 $\mathbb{E}[\tilde{A}'] \geq \sum_i c^{2i} n_i^2$, 所以，只需约束 F 的上界.

由于三角不等式，如果 $d(a, b) = \Delta$, 则对任何 $p \in V$ 必有 $d(p, a) \geq \frac{\Delta}{2}$ 或 $d(p, b) \geq \frac{\Delta}{2}$, 即每个顶点对应至少一条临边充分长. 设最大的非空区间下标为 $k = \log_c \Delta$, 则有 $k - \log_c 2 = \log_c \frac{\Delta}{2}$. 根据 I_j 的定义，有 $\sum_{k - \log_c 2 \leq j \leq k} n_j \geq n$. 由鸽巢原理，存在 $k - \log_c 2 \leq j \leq k$, 使得 $n_j \geq \frac{n}{\log_c 2}$. 令 $P = \{i : N_i \geq t := \alpha n\} - j$, 对应权重充分大的边组成的集合，注意这里的 α 是一个待优化的参数. 将 F 记为 $\frac{N_1 + N_2}{M_1 + M_2}$, 而 $M_1 = \sum_{i \in P} c^{2i} n_i^2$, $M_2 = \sum_{i \notin P} c^{2i} n_i^2$, $N_1 = \sum_{i \in P} c^{2i} n_i$, $N_2 = \sum_{i \notin P} c^{2i} n_i$. 由定义， $\frac{N_1}{M_1} \leq \frac{1}{t}$.

$$N_2 \leq t \sum c^{2i} \leq t \frac{c^{2(k+1)}}{c^2 - 1} \leq \frac{\Delta^2 (1 + \epsilon)^2}{\epsilon} t$$

而 $M_2 \geq (\frac{\Delta}{2} \frac{n}{\log_c^2 2})^2$, 故而

$$\frac{N_2}{M_2} \leq \frac{1}{n} \frac{4 \log_c^2 2 \alpha (1 + \epsilon)^2}{\epsilon}.$$

从而 $F \leq \max(\frac{N_1}{M_1}, \frac{N_2}{M_2})$. 设置 $\alpha = \Theta(\epsilon^{\frac{3}{2}})$, 得到 $F = O(\epsilon - \frac{3}{2} \frac{1}{n})$. 接下来只需根据式(1)调整算法输出坏结果的概率: 设置 $\epsilon = \Theta(\delta)$, $a = O(\delta^{-\frac{7}{2}})$, 我们得到了一个**常数概率下, 时间复杂度为 $O(\frac{n}{\delta^{\frac{7}{2}}})$ 的 $(1 + \delta)$ -近似算法。**

References

- [1] P. Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 428–434, 1999.