大数据算法-2025 春

Lecture 11: 1-median Problem 的次线性算法

2025.4.5

Lecturer: 丁虎 *Scribe:* 王向禄

Sublinear Algorithm（次线性算法）是一类算法 [3]，其时间复杂度或空间复杂度小于线性时间 $O(n)$ 或线性空间 $O(n)$（其中 n 是输入大小）。这些算法在处理大规模数据时尤为重要，因为它们可以避免遍历整个输入，节省计算资源和存储空间。

次线性算法的特点

- 时间复杂度小于 $O(n)$: 如 $O(logn)$、n^{ϵ} （其中 $0 < \epsilon < 1$）、$O(1)$ 等。
- 不需要完整读取输入：往往通过随机采样、哈希或其他近似方法，从部分数据中推断整体情况。
- 适用于大规模数据：如流数据处理、机器学习、图分析等领域。

1 1-median problem

"1-median problem" [1]（也叫 1-中位数问题或 1 median problem）是**设施选址问题**中的一个经典优化问题。它在运筹学、图论和数据科学中都有广泛应用。

Definition 1.1 (1-median problem)**.** 给定一个加权无向图 $G = (V, E)$，其中边的权重 $w(e)$ 满足三角不等式（即 G 是一个度量图），顶点数 $|V| = n$，边数 $|E| = \binom{n}{2}$ （即 G 是一个完全图）。目标是寻找一个顶点 $v^* \in V$ 使得下式最小化：

$$S(v) = \sum_{p \in V} w(p, v)$$

即选取一个顶点，使其到所有其他点的加权距离总和最小。

Goal: 找到一个顶点 $v_0 \in V$，使得：

$$S(v_0) \leq (1 + \delta) \min_{v \in V} S(v), \quad \delta > 0$$

想象一个 Oracle: 记作 $\Gamma_\delta(p, q)$

$$\begin{cases} S(p) > (1+\delta)S(q) & \text{返回}q \\ S(q) > (1+\delta)S(p) & \text{返回}p \\ S(p) \le (1+\delta)S(q) \text{ 且} S(q) \le (1+\delta)S(p) & \text{返回}p \text{ 或}q \end{cases}$$

那么做 $n-1$ 次比较就能得到一个近似最小的顶点（证明略）。接下来就是怎么构造上面这样的一个 Oracle，且该 Oracle 的时间复杂度是常数级别 $\Theta(1)$.

构造方法： 分别以 p, q 为顶点，r 为半径画球，定义 $H = \text{Ball}(p, r) \cup \text{Ball}(q, r)$，$t = w(q, p)$，$r = \frac{1}{\delta}t$。那么分类讨论

(1) 对于 $\forall v \notin H$，都有

- $|w(v,p) - w(v,q)| \le t$
- $w(v,p), w(v,q) \ge \frac{1}{\delta}t$
- $\frac{w(v,q)}{w(v,p)}, \frac{w(v,p)}{w(v,q)} \le \frac{w(v,p)+t}{w(v,q)} = 1 + \frac{t}{w(v,q)} \le 1+\delta$

所以所有球外的顶点都不考虑。

(2) 对于 $\forall v \in H$，都有

- $0 \le w(v,p) \le t + r = (1+\frac{1}{\delta})t$
- $0 \le w(v,q) \le t + r = (1+\frac{1}{\delta})t$

根据 Hoeffding Bounds。随机采样 m 个顶点（球内），对应的到顶点 p 的距离为 $x_1, x_2, \ldots, x_m$。如果我们希望

$$\text{Prob}[|\frac{1}{m}\sum_{i=1}^{m} x_i - \frac{1}{|H|}\sum_{v \in H} w(p,v)| \ge \epsilon(1+\frac{1}{\delta})t] \le e^{O(m\epsilon^2)} = O(\frac{1}{n})$$

那么 $m = O(\frac{1}{\epsilon^2} logn)$。因此只需要采样 m 个顶点就能近似 $\frac{1}{|H|}\sum_{v\in H} w(p,v)$，使得

$$\frac{1}{m}\sum_{i=1}^{m} x_i \in \bar{x} \pm \epsilon(1+\frac{1}{\delta})t$$

$$\frac{1}{m}\sum_{i=1}^{m} y_i \in \bar{y} \pm \epsilon(1+\frac{1}{\delta})t$$

其中 $\bar{x} = \frac{1}{|H|}\sum_{v\in H} w(p,v), \bar{y} = \frac{1}{|H|}\sum_{v\in H} w(q,v)$。

区分 $\bar{x}$ 和 $\bar{y}$: 假设 $\bar{x} > \bar{y}$, 那么有

$$\bar{x} - \epsilon(1+\frac{1}{\delta})t > \bar{y} + \epsilon(1+\frac{1}{\delta})t$$
$$\Rightarrow \epsilon < \frac{\bar{x}-\bar{y}}{2(1+\frac{1}{\delta})t}$$

根据三角不等式有 $\bar{x}+\bar{y} \geq t$，对 $\bar{x}$ 和 $\bar{y}$ 的取值范围进行分类讨论：

- (1) $\bar{x}, \bar{y} \geq \frac{1}{3}t$
- (2) $\bar{y} \leq \frac{1}{3}t, \bar{x} \geq \frac{2}{3}t$

得：

$$m = \Theta(\frac{1}{\delta^4} logn)$$

所以，在 $\Theta(\frac{1}{\delta^4} nlogn)$ 找到一个顶点 $v_0 \in V$，使得：

$$S(v_0) \leq (1+\delta)\min_{v\in V} S(v), \quad \delta > 0.$$

2 k-median Problem

Definition 2.1 (k-median)**.** 在集合 X 中寻找 k 个中心点 $c_1, \ldots, c_k \in X$，使得以下目标函数的值最小：

$$\sum_{p\in X} \min_{i=1,\ldots,k} d(p, c_i),$$

其中 $d(p, c_i)$ 表示点 p 到中心 c_i 的距离。

我们给出了一个时间复杂度为 $O(n)$ 的 k-median 问题近似算法。该方法采用 Korupolu 等人提出的 (α, β)-近似算法（以下简称 KPR 算法）[2]。令 $s = a\sqrt{kn\log k}$，其中 $a > 1$ 的具体值稍后给出。算法流程如下：

1. 从 X 中不放回地随机采样 s 个点，构成集合 S；
2. 在 S 上运行 KPR 的 k-median 算法，输出集合 $C' = \{c'_1, \ldots, c'_{\beta k}\}$；
3. 对于每个点 $p \in X$，将其分配给集合 C' 中离它最近的点，令 $d(p, C')$ 表示该距离；
4. 选取使得 $d(p, C')$ 最大的 $m = b\frac{kn}{\delta}\log k$ 个点构成集合 M，其中 b 稍后确定；

5. 在 M 上运行 KPR 算法，得到输出中心集合 C''；

6. 输出 C' 和 C''。

我们设定 $a=\Theta\left(\frac{1}{\delta}\sqrt{\log\frac{1}{\delta}}\right)$，$b=\Theta\left(\frac{1}{\delta^2}\log\frac{1}{\delta}\right)$。算法的总运行时间为

$$O\left(k^3 n\log k/\delta^2\log\frac{1}{\delta}\right)$$

Theorem 2.2. 对于任意常数 $\delta>0$，上述算法以概率 $\Omega(\delta)$ 返回一个

$$((1+\delta)3(2+\alpha),2\beta)$$

近似解，用于 *k-median* 问题。

Proof. 证明过程详见 [1] 定理 1. □

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.

[2] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of algorithms*, 37(1):146–188, 2000.

[3] R. Rubinfeld and A. Shapira. Sublinear time algorithms. *SIAM Journal on Discrete Mathematics*, 25(4):1562–1588, 2011.